The information in this USER’S MANUAL has been carefully reviewed and is believed to be accurate. The vendor assumes no responsibility for any inaccuracies that may be contained in this document, makes no commitment to update or to keep current the information in this manual, or to notify any person organization of the updates. Please Note: For the most up-to-date version of this manual, please see our web site at www.supermicro.com.

Super Micro Computer, Inc. (“Supermicro”) reserves the right to make changes to the product described in this manual at any time and without notice. This product, including software, if any, and documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any medium or machine without prior written consent.

IN NO EVENT WILL SUPERMICRO BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, SPECULATIVE, OR CONSEQUENTIAL DAMAGES ARISING FROM THE USE OR INABILITY TO USE THIS PRODUCT OR DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN PARTICULAR, SUPERMICRO SHALL NOT HAVE LIABILITY FOR ANY HARDWARE, SOFTWARE, OR DATA STORED OR USED WITH THE PRODUCT, INCLUDING THE COSTS OF REPAIRING, REPLACING, INTEGRATING, INSTALLING, OR RECOVERING SUCH HARDWARE, SOFTWARE, OR DATA.

Any disputes arising between manufacturer and customer shall be governed by the laws of Santa Clara County in the State of California, USA. The State of California, County of Santa Clara shall be the exclusive venue for the resolution of any such disputes. Super Micro's total liability for all claims will not exceed the price paid for the hardware product.

Information in this document is subject to change without notice. Other products and companies referred to herein are trademarks or registered trademarks of their respective companies or mark holders.

Copyright © 2020 by Super Micro Computer, Inc.

All rights reserved.

Printed in the United States of America

Manual Revision 2.0c

Release Date: June 3, 2020

Unless you request and receive written permission from Super Micro Computer, Inc., you may not copy any part of this document.
## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Rev.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 10, 2015</td>
<td>1.0</td>
<td>Created document.</td>
</tr>
<tr>
<td>October 5, 2015</td>
<td>1.0a</td>
<td>Minor formatting modifications.</td>
</tr>
<tr>
<td>June 5, 2017</td>
<td>1.0b</td>
<td>Added more APIs to section 2.3. Added and modified list of OEM APIs (Section 3.6). Added content to Section 3.7. Modified screenshots in Chapter 4. Modified reference links in Chapter 5.</td>
</tr>
<tr>
<td>March 20, 2018</td>
<td>2.0</td>
<td>Added Section 4 (Update service). Modified Section 3.6 (OEM APIs). Added new APIs in Section 2.3. Added new examples/screenshots in Section 5.</td>
</tr>
<tr>
<td>February 7, 2019</td>
<td>2.0a</td>
<td>Formatting modifications. Updated content in Section 2.3. Updated content in Section 3. Moved content from Section 3 to Section 5. Added and deleted content in Section 4. Added content to Section 5.</td>
</tr>
</tbody>
</table>
| November 19, 2019| 2.0b | Formatting modifications. Added section 5 - Secureboot. Added section 6.19 – KCS channel Control. Updated section 2.3 – Added the following new APIs:  
  - /redfish/v1/Managers/1/KCSInterface  
  - /redfish/v1/Chassis/MRVL-HA-RAID.[controller_num].StorageModule  
  - /redfish/v1/Systems/1/Storage/MRVL-HA-RAID  
  - /redfish/v1/Systems/1/Storage/MRVL-HA-RAID/Volumes  
  - /redfish/v1/Systems/1/Storage/MRVL-HA-RAID/Actions/Oem/Storage.CreateVD  
Updated section 3.2 – User Lockout configuration. Updated section 6.2 – Added Boot Order Configuration for System BIOS. |
<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 3, 2020</td>
<td>2.0c</td>
<td><strong>Updated section 6.4</strong> – Added example for Marvel SE9230 RAID configuration</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Updated section 2.3</strong> – Added Redfish 1.8 supported APIs to API list</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Added section 7.23</strong> – Certificate Service</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Added section 7.24</strong> – Virtual Media example for Redfish 1.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Updated section 6</strong></td>
</tr>
</tbody>
</table>
Revision History

1 Introduction

2 HTTP Request Methods

2.1 Responses

2.2 HTTP Status Code Description

2.3 List of Available APIs

3 Using RESTful APIs

3.1 Authentication

3.2 Account Service

3.3 Event Service

3.4 Registries

3.5 Jsonschema

4 UpdateService

4.1 Update SSL Certificate and Key

4.2 BIOS Update

4.3 BMC Firmware Update

4.4 Simple Update

5 Secure Boot

5.1 To Enable Redfish Secure Boot Refer to Below APIs

6 Device Management

6.1 NIC Device

6.2 GPU

6.3 NVMeSSD
6.4 PCIe Functions

7 Examples

7.1 System Reset

7.2 Configure the Boot Order in System BIOS

7.3 BIOS Configurations: Configure BIOS over Redfish

7.4 RAID Management Reference Examples

7.5 SMTP

A: SMTP SSL authentication Disabled

B: SMTP SSL authentication Enabled:

7.6 FanMode

7.7 Active Directory

7.8 Get/Set iKVM Mouse Mode

7.9 NTP

7.10 RADIUS

7.11 LDAP

7.12 Snooping

6.13 IP Access Control

7.14 SMCRAPK

7.15 SNMP

7.16 Syslog

7.17 Chassis Intrusion

7.18 IKVM

7.19 KCS Channel Control

7.20 Acknowledge Event

7.21 Getting MAC Address from System NICs

7.22 Python Code for Redfish API Response

7.23 Certificate Service

7.23.1 Generating CSR

7.23.2 Replace Certificate

7.23.3 Rekey Certificate

7.24 Virtual Media

7.24.1 Mount and set the VM settings

7.24.2 Verify whether the ISO is mounted from Redfish command
7.24.3 Unmount the ISO 51
7.24.4 Verify whether the ISO is unmounted from Redfish command; the node should be removed 52

8 Reference Links 52
1 Introduction

The Redfish Scalable Platforms Management API ("Redfish") is a new interface that uses RESTful interface semantics to access data defined in a model format to perform out-of-band systems management. It is suitable for a wide range of servers, from stand-alone to rack mount and blade environments, but scales equally well for large scale cloud environments.

Redfish is a management standard which uses data model representation inside of a hypermedia RESTful interface. It is based on REST, which is why Redfish is easier to use and implement than many other solutions. Since it is model oriented, it is capable of expressing the relationships between components in modern systems as well as the semantics of the services and components within them. It is also easily extensible. By using a hypermedia approach to REST, Redfish can express a large variety of systems from multiple vendors. Utilizing JSON (JavaScript Object Notation) data format, which is in plain text, allows many types of parameters to be available such that it enables scalability, human readability, and flexibility for most programming environments by easily interpreting payload.

The model is displayed in terms of an interoperable OData Schema with the payload of the messages being expressed in JSON following OData JSON conventions. The schema (available in both XML and JSON formats) includes annotations to facilitate the automatic translation of the schema to JSON Schema. The ability to externally host the schema definition of the resources in a machine-readable format allows the metadata to be associated with the data without encumbering Redfish services with the metadata, thus enabling more advanced client scenarios as found in many data center and cloud environments.

Supermicro enables Redfish feature sets on their X10/X11 platforms with 3.xx and 1.xx BMC firmware respectively. These features are covered under SFT-OOB-LIC and SFT-DCMS-SINGLE license. This document will provide you with an overview of Restful API services and describe how to receive Redfish API responses directly from a Supermicro BMC (Baseboard Management Controller).

2 HTTP Request Methods

The following HTTP methods are used to implement different actions, as described below.

- **Read Requests (GET):**
  The GET method is used to request a representation of a specified resource. The representation can be either a single resource or a collection.
- **Update (PATCH):**
  The PATCH method is used to apply partial modifications to a resource.
- **Replace (PUT):**
  The PUT method is used to completely replace a resource. Any properties omitted from the body of the request are reset to their default value.
• Create (POST):
The POST method is used to create a new resource. This request is submitted to the resource collection in which the new resource is meant to belong.
• Actions (POST):
The POST method may also be used to initiate operations on the object (Actions). The POST operation may not be idempotent.
• Delete (DELETE):
The DELETE method is used to remove a resource.

2.1 Responses
Four types of responses are supported, as defined below.

• Metadata Responses:
These describe the resources and types exposed by the service to generic clients.
• Resource Responses:
JSON representation of an individual resource.
• Resource Collection Responses:
JSON representation of a collection of resources.
• Error Responses:
Top-level JSON response providing additional information in the case of an HTTP error.

2.2 HTTP Status Code Description

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK</td>
</tr>
<tr>
<td>201</td>
<td>Created</td>
</tr>
<tr>
<td>202</td>
<td>Accepted</td>
</tr>
<tr>
<td>204</td>
<td>No Content</td>
</tr>
<tr>
<td>301</td>
<td>Moved Permanently</td>
</tr>
<tr>
<td>302</td>
<td>Found</td>
</tr>
<tr>
<td>304</td>
<td>Not Modified</td>
</tr>
<tr>
<td>400</td>
<td>Bad Request</td>
</tr>
<tr>
<td>401</td>
<td>Unauthorized</td>
</tr>
<tr>
<td>403</td>
<td>Forbidden</td>
</tr>
<tr>
<td>404</td>
<td>Not Found</td>
</tr>
<tr>
<td>405</td>
<td>Method Not Allowed</td>
</tr>
<tr>
<td>406</td>
<td>Not Acceptable</td>
</tr>
<tr>
<td>409</td>
<td>Conflict</td>
</tr>
<tr>
<td>410</td>
<td>Gone</td>
</tr>
<tr>
<td>411</td>
<td>Length Required</td>
</tr>
<tr>
<td>412</td>
<td>Precondition Failed</td>
</tr>
<tr>
<td>415</td>
<td>Unsupported Media Type</td>
</tr>
<tr>
<td>500</td>
<td>Internal Server Error</td>
</tr>
</tbody>
</table>
### 2.3 List of Available APIs

<table>
<thead>
<tr>
<th>API List</th>
<th>Notes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>/redfish/v1</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/SessionService</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Chassis</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/AccountService</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/EventService</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/UpdateService</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/TaskService</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Registries</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/JsonSchemas</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/SessionService/Sessions</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/SessionService/Sessions/[session_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/Thermal</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/Power</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/Sensors</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/Sensors/[sensor_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/NetworkAdapters</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/NetworkAdapters/[adapter_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/NetworkAdapters/[adapter_num]/NetworkPorts</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/NetworkAdapters/[adapter_num]/NetworkPorts/[np_num]</td>
<td></td>
</tr>
<tr>
<td>redfish/v1/Chassis/1/NetworkAdapters/[adapter_num]/NetworkDeviceFunctions</td>
<td></td>
</tr>
<tr>
<td>redfish/v1/Chassis/1/NetworkAdapters/[adapter_num]/NetworkDeviceFunctions/[ndf_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/PCIeSlots</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/PCIeDevices</td>
<td>Asset information of AOC NIC cards</td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/PCIeDevices/NIC[aoc_card_num]</td>
<td>Asset information of AOC NIC cards</td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/PCIeDevices/NIC[aoc_card_num]/PCleFunctions/[port_num]</td>
<td>Asset information of AOC NIC cards</td>
</tr>
<tr>
<td>URL</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/PCIeDevices/GPU[gpu_card_num]</td>
<td>Asset information of AOC NIC cards</td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/PCIeDevices/GPU[gpu_card_num]/PCIeFunctions/[gpu_instance_num]</td>
<td>Asset information of AOC NIC cards</td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/PCIeDevices/NVMeSSD[disk_num]</td>
<td>Asset information of AOC NIC cards</td>
</tr>
<tr>
<td>/redfish/v1/Chassis/1/PCIeDevices/PCIeSSD[disk_num]</td>
<td>Asset information of AOC NIC cards</td>
</tr>
<tr>
<td>/redfish/v1/Chassis/HA-RAID.[controller_num].StorageEnclosure.[enclosure_num]</td>
<td>For LSI 3108</td>
</tr>
<tr>
<td>/redfish/v1/Chassis/HBA.[controller_num].StorageEnclosure.[enclosure_num]</td>
<td>For LSI 3008</td>
</tr>
<tr>
<td>/redfish/v1/Chassis/HBA.[controller_num].StorageEnclosure.[enclosure_num]</td>
<td>For LSI 3008</td>
</tr>
<tr>
<td>/redfish/v1/Chassis/StorageBackplane</td>
<td>For PCH SATA or RSTe, TAS must be running</td>
</tr>
<tr>
<td>/redfish/v1/Chassis/StorageBackplane/Drives/Disk.Bay.[disk_num]</td>
<td>For PCH SATA or RSTe, TAS must be running</td>
</tr>
<tr>
<td>/redfish/v1/Chassis/NVMeSSD.[pcie_controller_num].Group.[group_num].StorageBackplane</td>
<td>For NVMe</td>
</tr>
<tr>
<td>/redfish/v1/Chassis/MRVL.HA-RAID.[controller_num].StorageModule</td>
<td>For SE9230</td>
</tr>
<tr>
<td>/redfish/v1/AccountService/Roles</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/AccountService/Roles/Administrator</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/AccountService/Roles/Operator</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/AccountService/Roles/ReadOnly</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/AccountService/Accounts</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/AccountService/Accounts/[account_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/Actions/Manager.Reset</td>
<td>BMC cold reset</td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/Actions/Oem/ManagerConfig.Reset</td>
<td>BMC factory default</td>
</tr>
<tr>
<td></td>
<td>&quot;Option&quot;: &quot;PreserveUser&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Option&quot;: &quot;ClearConfig&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Option&quot;: &quot;ResetToADMIN&quot;</td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/SerialInterfaces</td>
<td></td>
</tr>
</tbody>
</table>

Supermicro Redfish Reference Guide 11
<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/redfish/v1/Managers/1/SerialInterfaces/[service_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/NetworkProtocol</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/NetworkProtocol/HTTPS/Certificates</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/NetworkProtocol/HTTPS/Certificates/[cert_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/LogServices</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/LogServices/Log1</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/LogServices/Log1/Actions/LogService.ClearLog</td>
<td>Clear Log service</td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/LogServices/Log1/Entries</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/LogServices/Log1/Entries/[log_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/VM1</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/VM1/CfgCD</td>
<td>Configure ISO image settings: host, path, username/pass</td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/VM1/CfgCD/Actions/IsoConfig.Mount</td>
<td>Mount ISO image</td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/VM1/CfgCD/Actions/IsoConfig.UnMount</td>
<td>Unmount ISO image</td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/VM1/CD[mounted_dev_num]</td>
<td>User must first mount image</td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/VM1/Floppy[mounted_dev_num]</td>
<td>User must first mount image</td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/VM1/USB[mounted_dev_num]</td>
<td>User must first mount image</td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/VirtualMedia</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/VirtualMedia/CD[mounted_dev_num]</td>
<td></td>
</tr>
<tr>
<td>redfish/v1/Managers/1/VirtualMedia/CD[mounted_dev_num]/Actions/VirtualMedia.InsertMedia</td>
<td>Insert Media action</td>
</tr>
<tr>
<td>redfish/v1/Managers/1/VirtualMedia/CD[mounted_dev_num]/Actions/VirtualMedia.EjectMedia</td>
<td>Eject Media action</td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/VirtualMedia/Floppy[mounted_dev_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/VirtualMedia/USB[mounted_dev_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/EthernetInterfaces</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/EthernetInterfaces/[eth_num]</td>
<td></td>
</tr>
<tr>
<td>&lt;Managers OEM APIs&gt;</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/SNMP</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/SNMP/SNMPv2</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/SNMP/SNMPv3</td>
<td></td>
</tr>
<tr>
<td>URL Path</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/FanMode</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/MouseMode</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/Snooping</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/ActiveDirectory</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/ActiveDirectory/RoleGroups</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/ActiveDirectory/RoleGroups/[role_group]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/SMTP</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/Syslog</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/RADIUS</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/LDAP</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/SMCRAKP</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/IPAccessControl</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/IPAccessControl/FilterRule</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/IPAccessControl/FilterRule/[rule_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/NTP</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/IKVM</td>
<td>Get a URL link to launch iKVM/HTML5</td>
</tr>
<tr>
<td>/redfish/v1/Managers/1/KCSInterface</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems/1</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/Actions/ComputerSystem.Reset</td>
<td>System reset</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/Processors</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/Processors/[processor_num]</td>
<td></td>
</tr>
<tr>
<td>redfish/v1/Systems/1/ProcessorSummary/ProcessorMetrics</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/Memory</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/Memory/[memory_num]</td>
<td></td>
</tr>
<tr>
<td>redfish/v1/Systems/1/MemorySummary/MemoryMetrics</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/EthernetInterfaces</td>
<td>Data from BIOS and TAS</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/EthernetInterfaces/[eth_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/SimpleStorage</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/SimpleStorage/[controller_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/Storage</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/Storage/HA-RAID</td>
<td>For LSI 3108</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/Storage/HA-RAID/Volumes</td>
<td>For LSI 3108</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/Storage/HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]</td>
<td>For LSI 3108</td>
</tr>
<tr>
<td>URL</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>-------------</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]/Actions/Oem/Volume.Indicate</code></td>
<td>For LSI 3108; light on virtual drive indication LED: &quot;Active&quot;=&quot;true&quot;</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]/Actions/Oem/Volume.Delete</code></td>
<td>For LSI 3108; in logical view to delete specific virtual drive</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/HA-RAID/Actions/Oem/Storage.CreateVolume</code></td>
<td>For LSI 3108; create virtual drives</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/HA-RAID/Actions/Oem/Storage.ClearVolumes</code></td>
<td>For LSI 3108; in logical view to clear all configurations</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/HA-RAID/Actions/Oem/HARaidController.Save</code></td>
<td>For LSI 3108; save controller's &quot;BIOS Boot Mode&quot;</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/HBA</code></td>
<td>For LSI 3008</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/RAIDIntegrated</code></td>
<td>For RSTe, TAS must be running</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/RAIDIntegrated/Volumes</code></td>
<td>For RSTe, TAS must be running</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/RAIDIntegrated/Volumes/[volume_num]</code></td>
<td>For RSTe, TAS must be running</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/SATAEmbedded</code></td>
<td>For PCH SATA, TAS must be running</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/SATAEmbedded/Volumes</code></td>
<td>For PCH SATA, TAS must be running</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/SATAEmbedded/Volumes/[volume_num]</code></td>
<td>For PCH SATA, TAS must be running</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/MRVL.HA-RAID</code></td>
<td>For SE9230</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes</code></td>
<td>For SE9230</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]</code></td>
<td>For SE9230</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]/Actions/Oem/Volume.DeleteVD</code></td>
<td>For SE9230, Delete Virtual Drive</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]/Actions/Oem/Volume.RebuildVD</code></td>
<td>For SE9230, Rebuild Virtual Drive</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.[controller_num].Volume.[volume_num]/Actions/Oem/Volume.ImportVD</code></td>
<td>For SE9230, Import Virtual Drive</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Actions/Oem/Storage.CreateVD</code></td>
<td>For SE9230, Create Virtual Drive</td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/Bios</code></td>
<td>Current BIOS settings; only supports: X11DPT-B, X11DPT-PS, X11DPI, X11DPU, H11SSW,</td>
</tr>
<tr>
<td>Path</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/Bios/SD</td>
<td>Pending BIOS settings; only supports: X11DPT-B, X11DPT-PS, X11DPI, X11DPU, H11SSW, X11DDW, H12SSW, H12DST_B, X11DPU_Z, X11DPG_Q</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/Bios/Actions/Bios.ResetBios</td>
<td>Reset BIOS settings to default; only supports: X11DPT-B, X11DPT-PS, X11DPI, X11DPU, H11SSW, X11DDW, H12SSW, H12DST_B, X11DPU_Z, X11DPG_Q</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/SmcNodeManager</td>
<td>View SMC node manager policies</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/SmcNodeManager/Actions/SmcNodeManager.ClearAllPolicies</td>
<td>Clear SMC node manager policies</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/LogServices</td>
<td>System logs</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/LogServices/Log1</td>
<td>System logs</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/LogServices/Log1/Actions/LogService.ClearLog</td>
<td>Clear system management logs</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/LogServices/Log1/Actions/Oem/LogService.ClearAcknowledgements</td>
<td>Clear system log acknowledgements</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/LogServices/Log1/Entries</td>
<td>View system log entries</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/LogServices/Log1/Entries/[log_num]</td>
<td>Log entry details Patch to acknowledge</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/SecureBoot/</td>
<td>BIOS secureboot settings (Only X11DP supports)</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/SecureBoot/Actions/SecureBoot.ResetKeys</td>
<td>Reset key for secure boot (Only X11DP supports)</td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/BootOptions</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/BootOptions/[op_num]</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/NetworkInterfaces</td>
<td></td>
</tr>
<tr>
<td>/redfish/v1/Systems/1/NetworkInterfaces/[ni_num]</td>
<td></td>
</tr>
</tbody>
</table>

Supermicro Redfish Reference Guide 15
<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/redfish/v1/Systems/1/NetworkInterfaces/[ni_num]/NetworkPorts</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/Systems/1/NetworkInterfaces/[ni_num]/NetworkDeviceFunctions</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/EventService/Subscriptions</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/EventService/Subscriptions/[destination_num]</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/FirmwareInventory</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/FirmwareInventory/BMC</code></td>
<td>Only X11DP supports</td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/FirmwareInventory/BIOS</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/Actions/UpdateService.SimpleUpdate</code></td>
<td>Only X11DP supports</td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SimpleUpdateActionInfo</code></td>
<td>Only X11DP supports</td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/IPMIConfig</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/IPMIConfig/Actions/IPMIConfig.Upload</code></td>
<td>Upload new IPMI configuration file to set BMC</td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/IPMIConfig/Actions/IPMIConfig.Download</code></td>
<td>Download IPMI configuration as a file</td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SSLCert</code></td>
<td>View current SSL certification info</td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SSLCert/Actions/SSLCert.Upload</code></td>
<td>Used to upload new SSL certification file</td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SmcFirmwareInventory</code></td>
<td>Supported on X11 platforms</td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SmcFirmwareInventory/BMC</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.EnterUpdateMode</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Upload</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Update</code></td>
<td>&quot;PreserveCfg&quot;=&quot;true&quot;, &quot;PreserveSdr&quot;=&quot;true&quot;, &quot;PreserveSsl&quot;=&quot;true&quot;</td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Cancel</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.EnterUpdateMode</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Upload</code></td>
<td></td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Update</code></td>
<td>&quot;PreserveME&quot;=&quot;true&quot;, &quot;PreserveNVRAM&quot;=&quot;true&quot;, &quot;PreserveSMBIOS&quot;=&quot;true&quot;</td>
</tr>
<tr>
<td><code>/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Cancel</code></td>
<td>Only X11DP supports</td>
</tr>
</tbody>
</table>
**Revised API**

<table>
<thead>
<tr>
<th>Old</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>/redfish/v1/AccountService/Roles/Admin</td>
<td>/redfish/v1/AccountService/Roles/Administrator</td>
</tr>
<tr>
<td>/redfish/v1/AccountService/Roles/ReadOnlyUser</td>
<td>/redfish/v1/AccountService/Roles/Read-Only</td>
</tr>
<tr>
<td>/redfish/v1/UpdateService/FirmwareInventory</td>
<td>/redfish/v1/UpdateService/SmcFirmwareInventory</td>
</tr>
<tr>
<td>/redfish/v1/UpdateService/FirmwareInventory/BMC</td>
<td>/redfish/v1/UpdateService/SmcFirmwareInventory/BMC</td>
</tr>
<tr>
<td>/redfish/v1/UpdateService/FirmwareInventory/BMC/Actions/Oem/FirmwareInventory.EnterBMCUpdateMode</td>
<td>/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.EnterUpdateMode</td>
</tr>
<tr>
<td>/redfish/v1/UpdateService/FirmwareInventory/BMC/Actions/Oem/FirmwareInventory.UploadBMC</td>
<td>/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Upload</td>
</tr>
<tr>
<td>/redfish/v1/UpdateService/FirmwareInventory/BMC/Actions/Oem/FirmwareInventory.UpdateBMC</td>
<td>/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Update</td>
</tr>
<tr>
<td>/redfish/v1/UpdateService/FirmwareInventory/BMC/Actions/Oem/FirmwareInventory.CancelBMC</td>
<td>/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Cancel</td>
</tr>
<tr>
<td>/redfish/v1/UpdateService/FirmwareInventory/BIOS</td>
<td>/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS</td>
</tr>
<tr>
<td>/redfish/v1/UpdateService/FirmwareInventory/BIOS/Actions/Oem/FirmwareInventory.EnterBIOSUpdateMode</td>
<td>/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.EnterUpdateMode</td>
</tr>
<tr>
<td>/redfish/v1/UpdateService/FirmwareInventory/BIOS/Actions/Oem/FirmwareInventory.UploadBIOS</td>
<td>/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Upload</td>
</tr>
</tbody>
</table>
3 Using RESTful APIs

The user can receive API responses through programming by installing Postman or any other Rest API client application(s).

3.1 Authentication

Redfish supports both "Basic Authentication" and "Redfish Session Login Authentication" (as described below under Session Management). Service does not require a client to create a session when Basic Authentication is used.

3.1.1 Basic Authentication

HTTP BASIC authentication uses compliant TLS connections to transport the data between any third party authentication service and clients.

Note: Always check the status code once you get a response from the Redfish URL. You can refer to the status code table mentioned above. (All URLs/commands are case sensitive.)

3.1.2 Session Management

Redfish Service uses session management to implement authentication. This includes orphaned session timeouts and a number of simultaneous open sessions.

Step 1: The user can post the following username/password information in the payload field, which will create a new session.

```json
{
    "UserName": "<username>",
    "Password": "<password>
}
```

Example of applying for Authentication using a Chrome-based app (Advanced Rest Client): The user will receive 201 message code with X-AUTH token created.
• Users can create a maximum of 16 sessions.
• **Session lifetime:** For Redfish sessions, as long as a client sends requests for the session within the session timeout period, the session will remain open and the session authentication token will remain valid. If the session times out, the session will be automatically terminated.
• **According to Redfish specification, a user can define session time from 30s to 86400s.**
  If a user is not active in the defined time frame, then the token will be rendered invalid. Users can always patch “SessionTimeout” value if needed.
  Example: [PATCH] **https://BMC IP/redfish/v1/SessionService**
  Payload: {"SessionTimeout": 50}
• **Session termination or logout:** A Redfish session is terminated when the client logs out. This is accomplished by performing a DELETE to the session resource identified by the link returned in the location header either when the session was created or if the Session ID is returned in the response data. The ability to DELETE a session by specifying the session resource ID allows an administrator with sufficient privilege to terminate other users’ sessions from a different session.
  Example: [DELETE] **https://IP/redfish/v1/SessionService/Sessions/(num)** ->Send->Status Code: 200 OK

<table>
<thead>
<tr>
<th>Log in</th>
<th>Log out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation: POST</td>
<td>Operation: DELETE</td>
</tr>
<tr>
<td>URI: redfish/v1/SessionService/Sessions/</td>
<td>URI: redfish/v1/SessionService/Sessions/(num)</td>
</tr>
<tr>
<td>Request headers:</td>
<td>Request headers:</td>
</tr>
<tr>
<td>Content-Type: application/json</td>
<td>Content-Type: application/json</td>
</tr>
<tr>
<td>Request body:</td>
<td>Request body:</td>
</tr>
<tr>
<td>{&quot;UserName&quot;:&quot;UserName&quot;,&quot;Password&quot;:&quot;Password&quot;}</td>
<td>Requestbody: NONE</td>
</tr>
</tbody>
</table>
Step 2: The response will include an X-Auth-token header with a session token and a location header. Parse X-Auth token value to get an API response.
Note: The user can apply basic authentication as well.

3.2 Account Service
The user can perform the following operations under /redfish/v1/AccountService.
Methods supported: Get/Post/Patch/Delete

3.2.1 Create User
The user can create a new account using the following API and payload. The user can also delete respective accounts.

[POST] redfish/v1/AccountService/Accounts/
Payload:
{
"UserName":"User_Name",
"Password":"User_Password",
"RoleId":"role_id",  // Administrator, Operator, ReadOnlyUser
"Enabled":true
}
The user can also verify assigned privileges for different roles (ADMIN/Operator/Readonlyuser) under redfish/v1/AccountService/Roles.

3.2.2 User Lockout Configuration

[PATCH] redfish/v1/AccountService
Payload:
{"AuthFailureLoggingThreshold": 5,
"AccountLockoutThreshold": 2,
"AccountLockoutDuration": 300,
"AccountLockoutCounterResetAfter": 300}

3.3 Event Service
The event service is a new alert mechanism for Redfish. This alert will be sent out through HTTP or HTTPS to a web service that is subscribed to the service.
3.3.1 Add Subscription
Add a subscription to inform Redfish who will receive this event.

[POST]: https://IP/redfish/v1/EventService/Subscriptions/

Payload:
{"Destination":"http://www.dnsname.com/Destination1",
"Context":"user1_test",
"EventTypes":["Alert","StatusChange"],
"Protocol":"Redfish"}

Destination: Value shall contain a URI to the destination where the events will be sent.
Context: Value is a client-supplied string that is stored with the event destination subscription.
Protocol: This property shall contain the protocol type that the event will use for sending the event to the destination. A value of Redfish shall be used to indicate that the event type shall adhere to that defined in the Redfish specification.

EventTypes: Allowable values
- "StatusChange"
- "ResourceUpdated"
- "ResourceAdded"
- "ResourceRemoved"
- "Alert"

3.3.2 Test Event Subscription
Users can send a test event with “SendTestEvent” or generate an event in the BMC then Redfish will automatically send event alerts to subscriber(s).

[POST]: https://IP/redfish/v1/EventService/Actions/EventService.SendTestEvent

Payload:
{"EventType":"Alert"}

Users need to implement a RESTful event listener that can receive HTTP or HTTPS POST data that describes the Redfish event format. It can also subscribe to multiple services.

Refer to the Redfish-Event-Listener project page at GitHub to test Event Subscriptions or setup a Redfish Event Listener.

Sample data from Redfish Event Listener:
Time:Tue Feb 12 16:49:28 2019 Count:1
Host IP:('BMC_IP', 38486)
Event Details:{'@odata.context': '/redfish/v1/$metadata#EventService/Members/Events/58', '@odata.id': '/redfish/v1/EventService/Events/58', '@odata.type': '#EventService.v1_0_0.Event', 'Id': '58', 'Name': 'Event Array', 'Events': [{'EventType': 'Alert', 'Severity': 'OK', 'EventTimestamp': '2019/02/13'}}
00:49:04', 'Message': 'Submit Test Event', 'MessageArgs': ['/redfish/v1/EventService/Actions'], 'MessageId': '0', 'OriginOfCondition': {'@odata.id': '/redfish/v1/EventService'}, 'Context': 'Public'}

Time: Tue Feb 12 16:52:24 2019 Count: 2
Host IP: ('BMC_IP', 38500)
Event Details: {'@odata.context': '/redfish/v1/$metadata#EventService/Members/Events/59', '@odata.id': '/redfish/v1/EventService/Events/59', '@odata.type': '#EventService.v1_0_0.Event', 'Id': '59', 'Name': 'Event Array', 'Events': [{'EventType': 'Alert', 'Severity': 'Info', 'EventTimestamp': '2019/02/13 00:52:00', 'Message': 'Web login was successful.', 'MessageArgs': [], 'MessageId': 'Alert.1.0.LoginWeb', 'OriginOfCondition': {}, 'Context': 'Public'}]

3.3.3 View All Subscriptions
To see all subscriptions:
[GET]: https://IP/redfish/v1/EventService/Subscriptions/

3.3.4 Delete a Subscription
The user can delete subscription using the Delete request method.
[DELETE]: https://IP/redfish/v1/EventService/Subscriptions/(num)

3.4 Registries
/redfish/v1/Registries/Base.v1_4_0
Registry defines the base messages for Redfish. It represents properties for the registries themselves. The Message ID is formed per the Redfish specification. It consists of the RegistryPrefix concatenated with the version concatenated with the unique identifier for the message registry entry.

3.5 Jsonschema
/redfish/v1/JsonSchemas
The JSON Schema File resource describes the location (URI) of a particular Redfish schema definition being implemented or referenced by a Redfish service.

4 UpdateService

4.1 Update SSL Certificate and Key
Description: Update SSL certificate and key for secure web server connection.

[POST]: https://[BMC_IP]/redfish/v1/UpdateService/SSLCert/Actions/SSLCert.Upload
1. Change the type to “form-data”.
2. Select cert_file and key_file as keys and browse respective files to upload-> send.
4.2 BIOS Update

Description: Update BIOS through Redfish API. In the current implementation, the content type must be “multipart/form-data” while uploading the BIOS image.

4.2.1 Enter BIOS update mode by posting the following request and expect to receive a “Successfully Completed Request” response.

[POST]:
https://$BMC_IP/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.EnterUpdateMode

Note: The following screenshots are from the Restlet Chrome-based app.

4.2.2 Upload the BIOS image by posting the following request and expect to receive a “Successfully Completed Request” response. The content type must be “multipart/form-data”.

[POST]:
https://$BMC_IP/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Upload

4.2.3 Update BIOS by posting the following request with the following payload and expect to receive a “Successfully Completed Request” response.

[POST]:
https://$BMC_IP/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/Actions/SmcFirmwareInventory.Update

Payload:
{
    "PreserveME": true,
"PreserveNVRAM":true,
"PreserveSMBIOS":true }

Check the BIOS update status by issuing the following request with the GET method and expect to receive a response with the BIOS information.

[POST]: https://$BMC_IP/redfish/v1/UpdateService/SmcFirmwareInventory/BIOS/

4.3 BMC Firmware Update

Description: Update BMC firmware through the Redfish API. In the current implementation, the content type must be “multipart/form-data” while uploading the BMC image.

4.3.1 Enter BMC update mode by posting the following request and expect to receive a “Successfully Completed Request” response.
4.3.2 Upload the BMC image by issuing the following request with the POST method and expect to receive a “Successfully Completed Request” response. The content type must be “multipart/form-data”.

[POST]:
https://$BMC_IP/redfish/v1/UpdateService/SmcFirmwareInventory/BMC/Actions/SmcFirmwareInventory.Upload

Payload:
{
    "PreserveCfg":true,
    "PreserveSdr":true,
    "PreserveSsl":true
}
4.4 Simple Update

This action shall perform an update of installed software component(s) as contained within a software image file located at a URI referenced by the ImageURI parameter.

The user is required to prepare FTP, HTTP, or HTTPs file server to put BMC or BIOS firmware image file in.

[POST]: /redfish/v1/UpdateService/Actions/UpdateService.SimpleUpdate

Payload:
{
  "ImageURI": "<file ip>/<path and image file name>",
  "TransferProtocol": "FTP",
  "Targets": ["/redfish/v1/Managers/1"]
}

Target value:
For BIOS Update use "/redfish/v1/System/1"
For BMC Update use "/redfish/v1/Managers/1"
5 Secure Boot

UEFI Secure Boot was created to enhance security in the pre-boot environment. Secure Boot helps firmware, operating system, and hardware providers cooperate to thwart the efforts of malware developers.

Note: Please use supported BIOS for this function.
5.1 To Enable Redfish Secure Boot Refer to Below APIs

5.1.1 /redfish/v1/Systems/1/SafeBoot

Enable SecureBoot

[PATCH] /redfish/v1/Systems/1/SafeBoot
Payload:
{"SecureBoot":"Enabled"}

Check in Bios/SD.
Check in BIOS setup menu.

ResetKeysType

[POST] /redfish/v1/Systems/1/SafeBoot
{"ResetKeysType":"DeleteAllKeys"}

ResetKeysType Allowable Values:
"ResetAllKeysToDefault",
"DeleteAllKeys",
"DeletePK"

5.1.2 /redfish/v1/Systems/1/Bios

Set the 3 attributes below to BIOS to enable Secure Boot.

[POST] /redfish/v1/Systems/1/Bios
Payload:
{"SecureBoot":"Enabled",
"SecureBootMode":"User",
"ResetKeysType":"Delete PK Key"}
SecureBoot Allowable Values:
"Enabled",
"Disabled"

SecureBootMode allowable values:
"Setup",
"User",
"Audit",
"Deployed"

ResetKeyType allowable values:
"Disabled",
"Reset all keys to default",
"Delete all keys",
"Delete PK key"

6 Device Management

User can find details about all available network devices under /redfish/v1/Chassis/1/PCleDevices.

6.1 NIC Device

URI: /redfish/v1/Chassis/1/PCleDevices/NIC1/

Method supported: Get

Response:
6.2 GPU

URI: /redfish/v1/Chassis/1/PCIeDevices/GPU1
Method supported: Get
Response:

```json
{
    @odata.type:  "#PCIeDevice.v1_4_0.PCIeDevice",
    @odata.id:  "/redfish/v1/Chassis/1/PCIeDevices/GPU1",
    Id:  "GPU1",
    Name:  "GPU1",
    Description:  "GPU Device 1",
    Model:  "",
    SerialNumber:  "",
    PartNumber:  "",
    FirmwareVersion:  "",
    DeviceType:  "MultiFunction",
    Status:  {
        State:  "Enabled",
        Health:  "OK",
        HealthRollup:  "OK"
    },
    PCIeInterface:  {
        PCIeType:  "Gen3",
        MaxPCIeType:  "Gen3",
        LanesInUse:  18,
        MaxLanes:  16
    },
    PCIeFunctions:  {
        @odata.id:  "/redfish/v1/Chassis/1/PCIeDevices/GPU1/PCIeFunctions"
    },
    Links:  {
        Chassis:  {
            @odata.id:  "/redfish/v1/Chassis/1"
        }
    },
    Oem:  {
        Supermicro:  {
            @odata.type:  "#SecGPUExtensions.v1_0_0.GPU",
            GPUSlot:  1,
            Board part number:  "",
            Driver:  "Loaded",
            Memory vendor:  "",
            Memory part number:  "",
            GPU GUID:  "00000000000000000000000000000000",
            InfoROM version:  ""
        }
    }
}
```
6.3 NVMeSSD

URI: redfish/v1/Chassis/1/PCIeDevices/NVMeSSD1

Method supported: Get

Response:

```json
{
  "@odata.type": "#PCIeDevice.v1_4_0.PCIeDevice",
  "@odata.id": "/redfish/v1/Chassis/1/PCIeDevices/NVMeSSD1",
  "Id": "1",
  "Name": "NVMeSSD1",
  "Description": "NVMeSSD Device 1",
  "Manufacturer": "Intel",
  "Model": "INTEL SSDPE2UX50G7",
  "PartNumber": "INTEL SSDPE2UX50G7",
  "SerialNumber": "CYVF7162001JY589RH",
  "DeviceType": "Simulated",
  "Status": {
    "State": "Enabled",
    "Health": "OK",
    "HealthRollup": "OK"
  },
  "PCIeFunctions": {
    "@odata.id": "/redfish/v1/Chassis/1/PCIeDevices/NVMeSSD1/PCIeFunctions"
  },
  "links": {
    "Chassis": {
      "@odata.id": "/redfish/v1/Chassis/1"
    }
  },
  "Oem": {
    "Supermicro": {
      "Temperature": "38",
      "Port0MaxLinkSpeed": "8.0 GT/s",
      "Port0MaxLinkWidth": "4",
      "Port1MaxLinkSpeed": "N/A",
      "Port1MaxLinkWidth": "N/A",
      "InitialPowerRequirement": "10",
      "MaxPowerRequirement": "25"
    }
  }
}
```

6.4 PCIe Functions

URI: redfish/v1/Chassis/1/PCIeDevices/GPU1/PCIeFunctions/1

Method supported: Get

Response:
7 Examples

Users can integrate current APIs into their software and applications in order to receive all services provided by Redfish APIs.

7.1 System Reset

[POST]: https://$BMC_IP/redfish/v1/Systems/1/Actions/ComputerSystem.Reset

ResetType: AllowableValues

::{

7.2 Configure the Boot Order in System BIOS

Description: Change system boot order using Redfish.

1) **BootSourceOverrideEnabled**: Describes the state of the Boot Source Override feature.
   - **Disabled**: The system will boot normally.
   - **Once**: The system will boot (one time) to the Boot Source OverrideTarget.
   - **Continuous**: The system will boot to the target specified in the Boot SourceOverrideTarget until this property is set to Disabled.

2) **BootSourceOverrideMode**: The BIOS Boot Mode (either Legacy or UEFI) to be used when BootSourceOverrideTarget boot source is booted from.
   - **Legacy**: The system will boot in non-UEFI boot mode to the Boot Source Override Target.
   - **UEFI**: The system will boot in UEFI boot mode to the Boot Source Override Target.

3) **BootSourceOverrideTarget**: The current boot source to be used at next boot instead of the normal boot device if BootSourceOverrideEnabled is true.
Below values are allowed to set BootSourceOverrideTarget when BootSourceOverRideMode: UEFI

- PXE
- CD
- USB
- HDD
- UsbCd

when BootSourceOverRideMode: Legacy

- None
- PXE
- CD
- Floppy
- USB
- HDD
- BiosSetup
- UsbCd

Example: Change BootSourceOverrideTarget to BiosSetup.

[PATCH]: redfish/v1/Systems/1
{
  "Boot":
  "BootSourceOverrideEnabled": "Once",
  "BootSourceOverrideMode": "Legacy",
  "BootSourceOverrideTarget": "BiosSetup"
}

7.3 BIOS Configurations: Configure BIOS over Redfish

BIOS registry will show Menu of Key (Menus), Keys (Attributes), and Keys’ dependencies (Dependencies).

[GET]: https://$BMC_IP/redfish/v1/Registries/BiosAttributeRegistry.v1_0_0

Attributes: containing the attributes and their possible values.
Menu: containing the attributes menus and their hierarchy.

```json
{
  "DisplayName": "PCLe|PCI|PnP Configuration",
  "DisplayOrder": 26,
  "MenuPath": ".//Advanced/PCLe|PCI|PnPConfiguration",
  "Name": "PCLe|PCI|PnPConfiguration",
  "Hidden": false,
  "ReadOnly": false,
  "Value": []
}
```

Dependencies: a list of dependencies of attributes on this component.

```json
{
  "Dependency": {
    "MapFrom": {
      "MapFromAttribute": "PowerTechnology",
      "MapFromCondition": "NEQ",
      "MapFromProperty": "CurrentValue",
      "MapFromValue": "Custom",
      "MapTerms": "AND"
    },
    "MapToAttribute": "ENERGY PERF BIAS CFGmode",
    "MapToProperty": "GrayOut",
    "MapToValue": true,
    "DependencyType": "ENERGY PERF BIAS CFGmode",
    "Type": "Map"
  }
}
```

Example: If (PowerTechnology's CurrentValue != “Custom” AND PowerPerformanceTuning’s CurrentValue == “OS Controls EPB”) ENERGY_PERF_BIAS_CFGmode’s GrayOut = true

Modify attributes:
The user can GET current setting and PATCH desired settings.

[PATCH]: [https://$BMC_IP/redfish/v1/Systems/1/Bios](https://$BMC_IP/redfish/v1/Systems/1/Bios)
View pending settings:
The user can view any pending setting after PATCH. After PATCH, the user needs to reset the system to apply values to BIOS.

[GET]: https://$BMC_IP/redfish/v1/Systems/1/Bios/SD

**BIOS Reset:**
POST a reset of the BIOS attributes to default values. After POST, the user needs to reset the system to apply values to BIOS.
[POST]: https://$BMC_IP/redfish/v1/Systems/1/Bios/Actions/Bios.ResetBios"

**Change BIOS booting password:**
After POST, the user needs to reset the system to apply values to BIOS.

[POST]: https://$BMC_IP/redfish/v1/Systems/1/Bios/Actions/Bios.ChangePassword"
Payload:
"PasswordName":"AdminPassword"/"UserPassword",
"OldPassword":"
"NewPassword":"ADMIN"

Note: Please use supported BIOS for this function.
## 7.4 RAID Management Reference Examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
<th>URL</th>
<th>Method</th>
<th>Example Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create LSI3108 Volume</td>
<td>URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/HA-RAID/Actions/Oem/Storage.CreateVolume</td>
<td>Post</td>
<td>{ &quot;ControllerId&quot;:0, &quot;Raid&quot;: &quot;RAID0&quot;, &quot;Span&quot;: 1, &quot;PhysicalDrives&quot;:[&quot;HA-RAID.0.Disk.0&quot;, &quot;HA-RAID.0.Disk.1&quot;], &quot;UsePercentage&quot;:100, &quot;LogicalDriveCount&quot;:1, &quot;StripSizePerDDF&quot;:&quot;256K&quot;, &quot;LdReadPolicy&quot;:&quot;NoReadAhead&quot;, &quot;LdWritePolicy&quot;:&quot;WriteBack&quot;, &quot;LdIOPolicy&quot;:&quot;DirectIO&quot;, &quot;AccessPolicy&quot;:&quot;ReadWrite&quot;, &quot;DiskCachePolicy&quot;:&quot;Unchanged&quot;, &quot;InitState&quot;:&quot;NoInit&quot; }</td>
<td></td>
</tr>
<tr>
<td>Locate logical volume HDD</td>
<td>URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/HA-RAID/HA-RAID. [controller_num].Volumes/[volume_num]/Actions/OEM/Volume.Indicate</td>
<td>Post</td>
<td>{ &quot;Active&quot;:&quot;true&quot; }</td>
<td></td>
</tr>
<tr>
<td>Delete logical volume</td>
<td>URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/HA-RAID/HA-RAID. [controller_num].Volumes/[volume_num]/Actions/OEM/Volume.Delete</td>
<td>Post</td>
<td>{}</td>
<td></td>
</tr>
<tr>
<td>Clear all logical volumes</td>
<td>URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/HA-RAID/Storage.ClearVolumes</td>
<td>Post</td>
<td>{ &quot;ControllerId&quot;:0 }</td>
<td></td>
</tr>
<tr>
<td>Save HA-Raid controller config</td>
<td>URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/HA-RAID/Actions/Oem/HARAIDController.Save</td>
<td>Patch</td>
<td>{ &quot;ControllerId&quot;:0, &quot;BIOSBootMode&quot;:&quot;PauseOnError&quot;, &quot;JBODMode&quot;:&quot;Enable&quot; }</td>
<td></td>
</tr>
</tbody>
</table>
### View Marvell Drive details
URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/MRVL.HA-RAID
Method: Get

### Create Virtual Drive for Marvell
URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Actions/Oem/Storage.CreateVD
Method: Post
Example Body: {
"PD": ["MRVL.HA-RAID.0.StorageModule/Drives/Disk.Bay.0","MRVL.HA-RAID.0.StorageModule/Drives/Disk.Bay.1"],
"RaidLevel": "RAID1",
"StripeBlock": "64K",
"VDName": "SuperDrive"
}

### Delete Virtual Drive for Marvell
URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.0.Volume.0/Actions/Oem/Volume.DeleteVD
Method: Post
Example Body: {}

### Rebuild Virtual Drive for Marvell
URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.0.Volume.0/Actions/Oem/Volume.RebuildVD
Method: Post
Example Body: {}

### Import Virtual Drive for Marvell
Usage: Insert/import VD and register its UUID to Marvell firmware. Wait for the next system power-on, when this UUID will be registered to Marvell firmware.

URL: ${BMC_IP}/redfish/v1/Systems/1/Storage/MRVL.HA-RAID/Volumes/Controller.0.Volume.0/Actions/Oem/Volume.RebuildVD
Method: Post
Example Body: {}

## 7.5 SMTP
After applying the configurations, generate any system event to check if an email alert is received.

**A: SMTP SSL authentication Disabled:**

[PATCH]: redfish/v1/Managers/1/SMTP

Payload:

{"SmtpServer": "mailserver_ip or mailserver_name",
"SmtpPortNumber": "server_port",
"SmtpUserName": "",
"SmtpPassword": "",
"SmtpSenderAddress": "sender_email_address"}
B: SMTP SSL authentication Enabled:
[PATCH]: redfish/v1/Managers/1/SMTP
Payload:
{"SmtpSSLEnabled": true,
"SmtpServer": "mailserver_ip or mailserver_name",
"SmtpPortNumber": "server_port",
"SmtpUserName": "user_name",
"SmtpPassword": "user_password",
"SmtpSenderAddress": "sender_email_address"
}

7.6 FanMode

[PATCH]: redfish/v1/Managers/1/FanMode
Payload: ("Mode": "FullSpeed")
Mode Allowable Values:
{"Standard", "FullSpeed", "PUE2", "HeavyIO"}

7.7 Active Directory
The user can PATCH the following properties in order to configure ActiveDirectory.

[GET]: redfish/v1/Managers/1/ActiveDirectory
[POST]: "redfish/v1/Managers/1/ActiveDirectory/RoleGroups"
Payload:
{"RoleGroupName": "xxx",
"RoleGroupDomain": "xxx",
"RoleGroupPrivilege": "Operator"} // Administrator, Operator, User, NoAccess

[GET]/[PATCH]/[DELETE]: "redfish/v1/Managers/1/ActiveDirectory/RoleGroups/[number]"

Note: LDAP and Radius services need to be disabled first

7.8 Get/Set iKVM Mouse Mode
It is implemented under redfish/v1/Managers/1/MouseMode.
Methods supported: Get/Patch  
Allowable values: "Absolute", "Relative", "Single"

7.9 NTP  
Methods supported: Get/Patch  
[PATCH]: redfish/v1/Managers/1/NTP  
Payload:
{
"NTPEnable":true,
"PrimaryNTPServer":"127.0.0.1",
"SecondaryNTPServer":"localhost",
"DaylightSavingTime":false
}

7.10 RADIUS  
Methods supported: Get/Patch  
[PATCH]: redfish/v1/Managers/1/RADIUS.  
Payload:
{
"RadiusEnabled":true,
"RadiusServer":"127.0.0.1",
"RadiusPortNumber":1812,
"RadiusSecret":"SECRET"
}
Note: Active Directory and LDAP services need to be disabled first

7.11 LDAP  
Methods supported: Get/Patch  
[PATCH]: redfish/v1/Managers/1/LDAP  
Payload:
{
"LDAPEnabled":true,
"LDAPAuthOverSSL":true,
"LDAPPortNumber":389,
"LDAPServer":"0.0.0.0",
"LDAPPassword":"password",
"LDAPDN":"",
"LDAPSearchbase":""
}
Note: Active Directory and Radius services need to be disabled first

7.12 Snooping  
[GET]: https://$BMC_IP/redfish/v1/Managers/1/Snooping

6.13 IP Access Control  
It is implemented under redfish/v1/Managers/1/IPAccessControl.  
Methods supported: Get/Patch/Post
**7.14 SMCRAKP**
Methods supported: Get/Patch

[PATCH]: redfish/v1/Managers/1/SMCRAKP
Payload: \{"Mode":"Enabled"\}

**7.15 SNMP**
Methods supported: Get/Patch

[PATCH]: redfish/v1/Managers/1/SNMP
Payload: \{"SnmpEnabled":true\}

[PATCH]: redfish/v1/Managers/1/SNMPv2
Payload: \{"Snmpv2Enabled":true,"ROCommunity":"rtest","RWCommunity":"wtest"\}

[PATCH]: redfish/v1/Managers/1/SNMPv3
Payload: \{"Snmpv3Enabled":true,"UserName":"administrator","AuthProtocol":"SHA1","PrivateProtocol":"DES","AuthKey":"Test1234","PrivateKey":"Test1234"\}

**7.16 Syslog**
Methods supported: Get/Patch

[PATCH]: redfish/v1/Managers/1/Syslog
Payload: \{"EnableSyslog":true,"SyslogPortNumber":514,"SyslogServer":"10.136.176.16"\}

**7.17 Chassis Intrusion**
Methods supported: Get/Patch

[PATCH]: /redfish/v1/Chassis/1
Payload: \{"PhysicalSecurity":{"IntrusionSensor": "Normal"}\}

**7.18 IKVM**
Launch HTML5 iKVM using Redfish.

[GET]: {BMC_IP}/redfish/v1/Managers/1/IKVM
Use response property, “URI”, above to prepend “https://${BMC_IP}” and paste this complete URL in browser to render HTML5 iKVM.
Example of launching URL: https://[BMC_IP]/redfish/Kk1D4UVATDa0Jw.IKVM

7.19 KCS Channel Control
This feature allows the user to secure their environment by giving appropriate privilege to access the KCS interface.

[Administrator]: Any user accessing KCS interface will be able to do all the operations that the Administrator user can do.
[Operator]: Any user accessing the KCS interface will be able to do all the operations that a user with Operator privilege can do.
[User]: Any user accessing the KCS interface will be able to do all the operations that a user with User privilege can do.
[Callback]: This may be considered the lowest privilege level. Only commands necessary to support initiating a Callback are allowed.

[PATCH]: redfish/v1/Managers/1/KCSInterface
Payload: {"Privilege": "Administrator"}

7.20 Acknowledge Event

Description: Acknowledge event using Redfish.
It is implemented under redfish/v1/Systems/1/LogServices.
Methods supported: Get/Patch
- View events
  https://[BMC_IP]/redfish/v1/Systems/1/LogServices/Log1/Entries
• Acknowledge event: https://[BMC_IP]/redfish/v1/Systems/1/LogServices/Log1/Entries/1/ [PATCH]:
  {
    "Oem": {
      "Supermicro": {
        "MarkAsAcknowledged": true
      }
    }
  }

7.21 Getting MAC Address from System NICs
  https://[BMC_IP]/redfish/v1/Systems/1/EthernetInterfaces/1
7.22 Python Code for Redfish API Response
7.23 Certificate Service

The CertificateService describes a Certificate Service that represents the actions available to manage certificates and links to the certificates.

Method supported: Get

URL: redfish/v1/CertificateService/

Payload: {}

Response:

```
7.23.1 Generating CSR

Generate a certificate signing request (CSR) for the SSL certificate.
```
Generate CSR Action info

View the list of supported and required parameters to generate CSR.
Method supported: Get
URL: /redfish/v1/CertificateService/GenerateCSRActionInfo
Payload: {}
Response:

Generate CSR Request

This action is used to perform a certificate signing request.
Method supported: Post
URL: /redfish/v1/CertificateService/GenerateCSR
Payload: {
"Country": "US",
"State": "California",
"City": "San Jose",
"Organization": "Supermicro Computer",
"OrganizationalUnit": "PM",
"CommonName": "Supermicro.com",
"KeyPairAlgorithm": "TPM_ALG_RSA",
"CertificateCollection": {"@odata.id": "/redfish/v1/Managers/1/NetworkProtocol/HTTPS/Certificates"}
}
Response:
**View Certificate details**

**Method supported:** Get  
**URL:** /redfish/v1-Managers/1/NetworkProtocol/HTTPS/Certificates/1  
**Payload:** {}  
**Response:**

### 7.23.2 Replace Certificate

This action is used to replace an existing certificate.
**Replace Certificate Action info**

View the list of supported and required parameters to generate CSR.

Method supported: Get

URI: redfish/v1/CertificateService/ReplaceCertificateActionInfo

Payload: {}

Response:

**Replace Certificate**

Method supported: Post


Payload: {

"CertificateString": "-----BEGIN CERTIFICATE REQUEST-----
\nMIICvjCCAAwCAQAwEwEEMAQGA1UEBhMUeFJUETMBgNVVDAQFjJUTENBMQQx
nBgsqOgEh\n\nCertificationPath: CERTIFICATE
\n-----END CERTIFICATE REQUEST-----
"

"CertificateType": "PEM",

"CertificateUri": {

"@odata.id": "/redfish/v1/Managers/1/NetworkProtocol/HTTPS/Certificates/1"

}"}
7.23.3 Rekey Certificate
This action shall generate a new key pair for an existing certificate using the existing certificate data. The response shall contain a signing request that is to be signed by a certificate authority (CA). The service should retain the private key used for the generation of this request for when the certificate is installed. The private key should not be part of the response.

Method supported: Post
URI:
/redfish/v1/Managers/1/NetworkProtocol/HTTPS/Certificates/1/Actions/Certificate.Rekey
Payload: {
"KeyPairAlgorithm": "TPM_ALG_RSA"
}
Response:

![Response Image]
7.24 Virtual Media

7.24.1 Mount and set the VM settings

URL: ${BMC_IP}/redfish/v1/Managers/1/VirtualMedia/CD[mounted_dev_num]/Actions/VirtualMedia.InsertMedia

Method supported: Post

Body: {
    "Image": "<host>/<path>",
    "UserName": "some_username",
    "Password": "some_password"
}

POST https://172.31.57.27/redfish/v1/Managers/1/VirtualMedia/CD1/Actions/VirtualMedia.InsertMedia

```json
{
  "Success": {
    "code": "Base.v1_1_0.Success",
    "Message": "Successfully Completed Request. See ExtendedInfo for more information.",
    "@Message.ExtendedInfo": [
      {
        "MessageId": "SMC.v1.0.OemVmmMounted",
        "Severity": "Ok",
        "Resolution": "No resolution was required."
      }
    ]
  }
}
```
7.24.2 Verify whether the ISO is mounted from Redfish command
URL: `${BMC_IP}/redfish/v1/Managers/1/VirtualMedia/CD[mounted_dev_num]`
Method supported: Get
Body: LEAVE_IT_EMPTY

```json
{
  @odata.type: "#VirtualMedia.v1_3_2.VirtualMedia",
  @odata.id: "/redfish/v1/Managers/1/VirtualMedia/CD1",
  Id: "CD1",
  Name: "Virtual Removable Media",
  MediaTypes: ["CD", "DVD"],
  ImageName: "windows2016-6G.iso",
  ConnectedVia: "URI",
  Inserted: true,
  WriteProtected: true,
  TransferProtocolType: "HTTP",
  TransferMethod: "Stream",
}
```

7.24.3 Unmount the ISO
URL: `${BMC_IP}/redfish/v1/Managers/1/VirtualMedia/CD[mounted_dev_num]/Actions/VirtualMedia.EjectMedia`
Method supported: Post
Body: {}

```json
Success: {
  code: "Base.v1_4_0.Success",
  Message: "Successfully Completed Request. See ExtendedInfo for more information."
}
```
7.24.4 Verify whether the ISO is unmounted from Redfish command; the node should be removed

URL: ${BMC_IP}/redfish/v1/Managers/1/VirtualMedia/CD[mounted_dev_num]

Method supported: Get

Body: LEAVE_IT_EMPTY

```json
@odata.type: "#VirtualMedia.v1_3_2.VirtualMedia",
@odata.id: "/redfish/v1/Managers/1/VirtualMedia/CD1",
Id: "CD1",
Name: "Virtual Removable Media",
MediaTypes: ["CD", "DVD"],
ImageName: "windows2016-6G.iso",
ConnectedVia: "NotConnected",
Inserted: false,
WriteProtected: true,
TransferProtocolType: "HTTP",
TransferMethod: "Stream",
```

8 Reference Links

Supermicro Redfish:
https://www.supermicro.com/solutions/Redfish.cfm

Supermicro YouTube:
https://www.youtube.com/watch?v=anppU663kUs

DMTF Redfish:
http://www.dmtf.org/standards/redfish
http://redfish.dmtf.org/

Mockups:
http://redfish.dmtf.org/redfish/v1

Contact:
Supermicro Technical Support
support@supermicro.com