



Supermicro Redfish Plug-in for Nagios Core

User's Guide

Revision 1.0

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.

FCC Statement: This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the manufacturer's instruction manual, may cause harmful interference with radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case you will be required to correct the interference at your own expense.

California Best Management Practices Regulations for Perchlorate Materials: This Perchlorate warning applies only to products containing CR (Manganese Dioxide) Lithium coin cells. Perchlorate Material-special handling may apply. See <http://www.dtsc.ca.gov/hazardouswaste/perchlorate/> for further details.

Manual Revision 1.0

Release Date: 2018/10/31

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

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 © 2018 by Super Micro Computer, Inc.

All rights reserved.

Printed in the United States of America

Revision History

Date	Rev	Description
2018/10/31	1.0	Initial release

Contents

1. Overview	7
1.1 Prerequisites	8
1.1.1 System Requirements	8
1.1.2 Firmware Requirements for Remote Servers	8
1.1.3 Additional Package Requirements for Remote Servers	8
2. Installation	9
2.1 Installing on a Standalone	9
2.2 Installing the Plug-in for Nagios Core	9
3. Commands and Usage	11
3.1 Basic Usage	11
3.2 List of Commands	11
3.3 BIOS Configuration File in JSON Format	12
4. Managing Server	14
4.1 Health Information	14
4.1.1 Getting Health Information on the Entire System	15
4.1.2 Getting Health Information on the Fan Sensors	17
4.1.3 Getting Health Information on the Temperature Sensors	17
4.1.4 Getting Health Information on the Voltage Sensors	18
4.1.5 Getting Health Information on the Power Supplies	18
4.1.6 Getting Health Information on the Storage Component	19
4.1.7 Getting Performance Data	19
4.2 Hardware Information	20
4.2.1 Getting Hardware Information	21
4.3 BIOS Firmware Management	27
4.3.1 Listing BIOS Configurations	27
4.3.2 Exporting BIOS Configurations	28
4.3.3 Updating BIOS Configurations	28
4.3.4 Listing BIOS Firmware Information	29
4.3.5 Updating BIOS Firmware	29
4.4 BMC Firmware Management	31
4.4.1 Listing BMC Firmware Information	31

4.4.2	Updating BMC Firmware	31
4.4.3	BMC Cold Reboot.....	32
4.5	Event Service.....	33
4.5.1	Listing Subscriptions	33
4.5.2	Adding a Subscription.....	33
4.5.3	Deleting a Subscription.....	34
5.	Event Receiver	35
5.1	Supermicro Event Receiver.....	35
5.2	Event Log.....	35
	Contacting Supermicro	36

1. Overview

The Supermicro Redfish Plug-in provides a command line interface for remote management and monitoring of a Supermicro server via Redfish. The plug-in can integrate with Nagios Core to monitor server health.

Via this plug-in, health information on the following components is available:

- System Health
- Memory
- Fan sensors
- Temperature sensors
- Voltage sensors
- Power supply
- Storage

The management features include:

- Updating firmware for BIOS and BMC
- Updating BIOS configuration
- Hardware and firmware Inventory
- Managing event subscriptions

1.1 Prerequisites

1.1.1 System Requirements

Environment	Requirements
Operating System	CentOS 7.0 (x86_64) Ubuntu 16.04 (x86_64)

1.1.2 Firmware Requirements for Remote Servers

Environment	Requirements
BMC	X10 ATEN platform (REDFISH_X10): 3.77 or later. X11 ATEN platform (SMT_X11): 1.49 or later. X11 ATEN platform (SMT_X11AST2500): 1.55 or later. X11 ATEN platform (SMT_X11IPMI): 1.55 or later.
License	SFT-DCMS-Single



Notes:

- Latest BMC firmware can be downloaded from the Supermicro website at https://www.supermicro.com/support/resources/bios_ipmi.php.
-

1.1.3 Additional Package Requirements for Remote Servers

Features	Requirements
<ul style="list-style-type: none">● PCH SATA information● Intel RSTe information● Ethernet interface information	TAS_1.5.1_build.180202

2. Installation

2.1 Installing on a Standalone

1. Extract **nagios_supermicro_redfish_x.x.x_build.xxxxxx.tar.gz**
2. Execute **nagios_supermicro_redfish** in the **bin** directory

2.2 Installing the Plug-in for Nagios Core

1. Copy the executable file **nagios_supermicro_redfish** in the **bin** directory to the Nagios plug-in directory, e.g., `/usr/local/nagios/libexec`.
2. Add the command definition to the Nagios configuration file, e.g., `etc/objects/commands.cfg`.
 - Command definition:

```
$USER1$/nagios_supermicro_redfish -i $_HOSTIPMI_IP$ -u $_HOSTIPMI_USER$ -p $_HOSTIPMI_PWD$ $ARG1$
```

- Example:

```
define command{
    command_name    check_supermicro_redfish
    command_line    $USER1$/nagios_supermicro_redfish -i $_HOSTIPMI_IP$ -u $_HOSTIPMI_USER$ -p $_HOSTIPMI_PWD$ $ARG1$
}
```

3. Add the host definition.

The following attributes should be provided in the host definition.

Attribute	Description
<code>_ipmi_ip</code>	IPMI address of remote server
<code>_ipmi_user</code>	IPMI login username of remote server
<code>_ipmi_pwd</code>	IPMI login password of remote server

- Example

```
define host {
    use                linux-server
    host_name          x11
    alias              x11
    _ipmi_ip           10.136.160.176
    _ipmi_user         ADMIN
    _ipmi_pwd          ADMIN
}
```

4. Add the service definition

For the **check_command** definition, see [4.1 Health Information Command](#).

- Example

```
define service {  
    use                generic-service  
    host_name          x11  
    service_description supermicro redfish for system  
    check_command       check_supermicro_redfish!'check'  
}
```

5. Restart the Nagios service

3. Commands and Usage

3.1 Basic Usage

Usage:

```
nagios_supermicro_redfish [-h] [-d] [--version]
                          -i <ip> -u <username> -p <password>
                          [<command>] [<command option>]
```

Common Options

Option	Description
-i <ip>	Remote server IP address.
-u <username>	Remote server username.
-p <password>	Remote server password.
-h, --help	Shows help message.
--version	Shows version.
-d, --debug	Shows the debug message.

3.2 List of Commands

Health Information		
Command	Option	Description
check	--skip_storage (Optional) Skip to check the storage components.	Get health information of system and storage systems. It shows the failure components if system is in abnormal status by default.
	--show_all (Optional) Show all components	
	-t, --type <type> (Optional) Show specific type of components	
	<type> fan fan sensors temp temperature sensors volt voltage sensors ps power supplies storage storage components perf fan, temp sensors and power consumption	
Hardware Information		
Command	Option	Description
hwinfo	-f <file>, --file <file> (Optional) Export to file.	Get the hardware information.

BIOS Management		
Command	Option	Description
bios config list		List all BIOS configurations.
bios config export	-f <file>, --file <file> File location.	Export the current BIOS configurations to an assigned file.
bios config update	-f <file>, --file <file> File location.	Update BIOS configuration with given configuration file.
bios fw info		Get the BIOS firmware information.
bios fw update	-f <file>, --file <file> File location. --preserve_nv (Optional) Preserve NVRAM data. --preserve_mer (Optional) Preserve the ME firmware region.	Update BIOS firmware with given image file.
BMC Management		
Command	Option	Description
bmc fw info		Get the BMC firmware information.
bmc fw update	-f <file>, --file <file> File location. --overwrite_cfg (Optional) restore the BMC's factory default setting. --overwrite_sdr (Optional) restore the SDR defaults --overwrite_sslcert (Optional) restore the default SSL certificate	Update BMC firmware with given image file.
bmc reset		BMC cold reboot
Event Service		
Command	Option	Description
event-service list		List all subscriptions.
event-service add	dest URL of destination --type [EVENT_TYPE] (Optional) event type: StatusChange, ResourceUpdated, ResourceAdded, ResourceRemoved and Alert (default:StatusChange, Alert) --context [CONTEXT] (Optional) Context string (Default: "public")	Add subscription.
event-service del	id Subscription ID	Delete subscription ID.

3.3 BIOS Configuration File in JSON Format

The BIOS configuration file contains the BIOS attributes updated by users and each attribute is formatted as a JSON object.

The key definition in the JSON object:

Key	Description
-----	-------------

Name	Attribute name
Description	Attribute description
Value	Current value of attribute
ValueType	Value type of attribute
AllowableValue	The allowable value

If the **ValueType** is Boolean, the acceptable values are **false** and **true**. If the **ValueType** is Enumeration, the acceptable values are listed in **AllowableValue**.

Example:

```
{
  "Name": "QuietBoot",
  "Description": "Enables or disables Quiet Boot option",
  "Value": true,
  "ValueType": "Boolean",
  "AllowableValue": null
},
{
  "Name": "PowerButtonFunction",
  "Description": "Select the power button function.",
  "Value": "4 Seconds Override",
  "ValueType": "Enumeration",
  "AllowableValue": [
    {
      "ValueDisplayName": "Instant Off"
    },
    {
      "ValueDisplayName": "4 Seconds Override"
    }
  ]
}
```

4. Managing Server

4.1 Health Information

The detail health information of each type of components or sensors is summarized in the table below.

Component	Attributes
Computer system	<ul style="list-style-type: none">- health- state- model- serial number
Memory	<ul style="list-style-type: none">- health- state- manufacturer- part number- serial number
Temperature sensor	<ul style="list-style-type: none">- health- state- name- reading (C)- upper threshold critical (C)- lower threshold critical (C)
Fan sensor	<ul style="list-style-type: none">- health- state- name- reading (RPM)- upper threshold critical (RPM)- lower threshold critical (RPM)
Voltage sensor	<ul style="list-style-type: none">- health- state- name- reading (Volts)- upper threshold critical (Volts)- lower threshold critical (Volts)
Power supply	<ul style="list-style-type: none">- health- state- name- model
Storage	Storage System <ul style="list-style-type: none">- health- state Controller <ul style="list-style-type: none">- health- state- controller ID

	<ul style="list-style-type: none"> - controller name - manufacturer - model - enclosures Drive <ul style="list-style-type: none"> - health - state - chassis ID - drive name - manufacturer - model - serial number Volume <ul style="list-style-type: none"> - health - state - controller ID - volume name - type - capacity - drives
--	---

Four types of health status are defined by Nagios: **OK**, **WARNING**, **CRITICAL**, and **UNKNOWN**. The **state** is defined in Redfish which indicates the known state of the resource.

The **state** definition:

State	Description
Enabled	This function or resource has been enabled.
Disabled	This function or resource has been disabled.
StandbyOffline	This function or resource is enabled, but awaiting an external action to activate it.
StandbySpare	This function or resource is part of a redundancy set and is awaiting a failover or other external action to activate it.
InTest	This function or resource is undergoing testing.
Starting	This function or resource is starting.
Absent	This function or resource is not present or not detected.
UnavailableOffline	This function or resource is present but cannot be used.
Deferring	The element will not process any commands but will queue new requests.
Quiesced	The element is enabled but only processes a restricted set of commands.
Updating	The element is updating and may be unavailable or degraded.

4.1.1 Getting Health Information on the Entire System

Using “check” command to get the overall status of the computer system and storage systems. The status of storages systems are not associated with computer system.

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password>  
check [--skip_storage][--show_all]
```

Example:

```
$. /nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN check
```

Console Output:

```
CRITICAL - System, State=Enabled, Model=X11DGQ, S/N=  
CRITICAL - Power Supply Bay 1, State=Enabled  
OK - SATAEmbedded:Embedded SATA Storage System, State=Enabled
```

Example: Show all components

```
$. /nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN check --show_all
```

Console Output:

```
CRITICAL - System, State=Enabled, Model=X11DGQ, S/N=  
OK - P1-DIMMA1, State=Enabled, Manufacturer=Micron, P/N=16ATF1G64AZ-2G1A1, S/N=0E400B6B, Capacity=8 GB  
N/A - FAN1, State=Absent  
N/A - FAN2, State=Absent  
N/A - FAN3, State=Absent  
N/A - FAN4, State=Absent  
N/A - FAN5, State=Absent  
N/A - FAN6, State=Absent  
N/A - FAN7, State=Absent  
OK - FAN8, State=Enabled, Reading=700RPM, LowLimit=500, HighLimit=25400  
OK - FAN9, State=Enabled, Reading=1400RPM, LowLimit=500, HighLimit=25400  
OK - CPU1 Temp, State=Enabled, Reading=49C, LowLimit=0, HighLimit=102  
N/A - CPU2 Temp, State=Absent  
OK - PCH Temp, State=Enabled, Reading=35C, LowLimit=0, HighLimit=85  
N/A - Inlet Temp, State=Absent  
OK - System Temp, State=Enabled, Reading=27C, LowLimit=0, HighLimit=85  
OK - Peripheral Temp, State=Enabled, Reading=33C, LowLimit=0, HighLimit=85  
OK - MB_10G Temp, State=Enabled, Reading=34C, LowLimit=0, HighLimit=100  
OK - VRMCpu1IN Temp, State=Enabled, Reading=36C, LowLimit=0, HighLimit=100  
OK - VRMCpu1IO Temp, State=Enabled, Reading=38C, LowLimit=0, HighLimit=100  
OK - VRMCpu2IN Temp, State=Enabled, Reading=29C, LowLimit=0, HighLimit=100  
OK - VRMCpu2IO Temp, State=Enabled, Reading=28C, LowLimit=0, HighLimit=100  
OK - VRMP1AB Temp, State=Enabled, Reading=35C, LowLimit=0, HighLimit=100  
OK - VRMP1DE Temp, State=Enabled, Reading=34C, LowLimit=0, HighLimit=100  
OK - VRMP2AB Temp, State=Enabled, Reading=32C, LowLimit=0, HighLimit=100  
OK - VRMP2DE Temp, State=Enabled, Reading=29C, LowLimit=0, HighLimit=100  
OK - P1-DIMMA1 Temp, State=Enabled, Reading=33C, LowLimit=0, HighLimit=85  
N/A - P1-DIMMB1 Temp, State=Absent  
N/A - P1-DIMMC1 Temp, State=Absent  
N/A - P1-DIMMD1 Temp, State=Absent  
N/A - P1-DIMME1 Temp, State=Absent  
N/A - P1-DIMMF1 Temp, State=Absent  
N/A - P2-DIMMA1 Temp, State=Absent  
N/A - P2-DIMMB1 Temp, State=Absent  
N/A - P2-DIMMC1 Temp, State=Absent  
N/A - P2-DIMMD1 Temp, State=Absent  
N/A - P2-DIMME1 Temp, State=Absent  
N/A - P2-DIMMF1 Temp, State=Absent  
N/A - M.2-H Temp, State=Absent  
OK - 12V, State=Enabled, Reading=12.096, LowLimit=10.296, HighLimit=13.236  
OK - 5VCC, State=Enabled, Reading=4.978, LowLimit=4.282, HighLimit=5.529  
OK - 3.3VCC, State=Enabled, Reading=3.384, LowLimit=2.823, HighLimit=3.656  
OK - Vcpu1, State=Enabled, Reading=1.771, LowLimit=1.258, HighLimit=2.086  
OK - Vcpu2, State=Absent, Reading=0, LowLimit=0, HighLimit=0  
OK - VdimmP1AB, State=Enabled, Reading=1.223, LowLimit=1.062, HighLimit=1.587  
OK - VdimmP1DE, State=Enabled, Reading=1.216, LowLimit=1.062, HighLimit=1.587  
OK - VdimmP2AB, State=Absent, Reading=0, LowLimit=0, HighLimit=0  
OK - VdimmP2DE, State=Absent, Reading=0, LowLimit=0, HighLimit=0  
OK - 5VSB, State=Enabled, Reading=5.05, LowLimit=4.27, HighLimit=5.53  
OK - 3.3VSB, State=Enabled, Reading=3.35, LowLimit=2.823, HighLimit=3.656  
OK - 1.8V PCH, State=Enabled, Reading=1.779, LowLimit=1.608, HighLimit=1.995  
OK - PVNN PCH, State=Enabled, Reading=1, LowLimit=0.778, HighLimit=1.108  
OK - 1.05V PCH, State=Enabled, Reading=1.062, LowLimit=0.894, HighLimit=1.342  
OK - 12VSB, State=Enabled, Reading=11.976, LowLimit=10.296, HighLimit=13.236  
CRITICAL - Power Supply Bay 1, State=Enabled
```

```

OK - Power Supply Bay 2, State=Enabled, Model=PWS-1K02A-1R
N/A - Power Supply Bay 3, State=Absent
N/A - Power Supply Bay 4, State=Absent
OK - Intrusion Sensor, State=Normal
OK - SATAEmbedded:Embedded SATA Storage System, State=Enabled
OK - SATAEmbedded.0:System SATA, State=Enabled, Manufacturer=None, Model=None, Enclosures=StorageBackplane
OK - SATAEmbedded#Volume0:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=500 MB,
    Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume1:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=115.71 GB,
    Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume2:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=1014 MB,
    Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume3:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=115.69 GB,
    Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume4:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=115.69 GB,
    Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume5:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=49.98 GB,
    Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume6:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=7.69 GB,
    Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume7:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=57.96 GB,
    Drives=[StorageBackplane#Disk.Bay.3]
OK - StorageBackplane#Disk.Bay.3, State=Unknown, S/N=9XE0HT8W

```

4.1.2 Getting Health Information on the Fan Sensors

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> check -t fan
```

Example:

```
$ ./nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN check -t fan
```

Console Output:

```

N/A - FAN1, State=Absent
N/A - FAN2, State=Absent
N/A - FAN3, State=Absent
N/A - FAN4, State=Absent
N/A - FAN5, State=Absent
N/A - FAN6, State=Absent
N/A - FAN7, State=Absent
OK - FAN8, State=Enabled, Reading=800RPM, LowLimit=500, HighLimit=25400
OK - FAN9, State=Enabled, Reading=1600RPM, LowLimit=500, HighLimit=25400

```

4.1.3 Getting Health Information on the Temperature Sensors

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> check -t temp
```

Example:

```
$ ./nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN check -t temp
```

Console Output:

```

OK - CPU1 Temp, State=Enabled, Reading=55C, LowLimit=0, HighLimit=102
N/A - CPU2 Temp, State=Absent
OK - PCH Temp, State=Enabled, Reading=36C, LowLimit=0, HighLimit=85
N/A - Inlet Temp, State=Absent
OK - System Temp, State=Enabled, Reading=28C, LowLimit=0, HighLimit=85
OK - Peripheral Temp, State=Enabled, Reading=33C, LowLimit=0, HighLimit=85
OK - MB_10G Temp, State=Enabled, Reading=34C, LowLimit=0, HighLimit=100
OK - VRMCpu1IN Temp, State=Enabled, Reading=37C, LowLimit=0, HighLimit=100
OK - VRMCpu1IO Temp, State=Enabled, Reading=39C, LowLimit=0, HighLimit=100
OK - VRMCpu2IN Temp, State=Enabled, Reading=29C, LowLimit=0, HighLimit=100

```

```
OK - VRMCpu2IO Temp, State=Enabled, Reading=28C, LowLimit=0, HighLimit=100
OK - VRMP1AB Temp, State=Enabled, Reading=36C, LowLimit=0, HighLimit=100
OK - VRMP1DE Temp, State=Enabled, Reading=34C, LowLimit=0, HighLimit=100
OK - VRMP2AB Temp, State=Enabled, Reading=32C, LowLimit=0, HighLimit=100
OK - VRMP2DE Temp, State=Enabled, Reading=29C, LowLimit=0, HighLimit=100
OK - P1-DIMMA1 Temp, State=Enabled, Reading=33C, LowLimit=0, HighLimit=85
N/A - P1-DIMMB1 Temp, State=Absent
N/A - P1-DIMMC1 Temp, State=Absent
N/A - P1-DIMMD1 Temp, State=Absent
N/A - P1-DIMME1 Temp, State=Absent
N/A - P1-DIMMF1 Temp, State=Absent
N/A - P2-DIMMA1 Temp, State=Absent
N/A - P2-DIMMB1 Temp, State=Absent
N/A - P2-DIMMC1 Temp, State=Absent
N/A - P2-DIMMD1 Temp, State=Absent
N/A - P2-DIMME1 Temp, State=Absent
N/A - P2-DIMMF1 Temp, State=Absent
N/A - M.2-H Temp, State=Absent
```

4.1.4 Getting Health Information on the Voltage Sensors

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> check -t volt
```

Example:

```
$ ./nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN check -t volt
```

Console Output:

```
OK - 12V, State=Enabled, Reading=11.916, LowLimit=10.296, HighLimit=13.236
OK - 5VCC, State=Enabled, Reading=4.862, LowLimit=4.282, HighLimit=5.529
OK - 3.3VCC, State=Enabled, Reading=3.316, LowLimit=2.823, HighLimit=3.656
OK - Vcpu1, State=Enabled, Reading=1.726, LowLimit=1.258, HighLimit=2.086
OK - Vcpu2, State=Absent, Reading=0, LowLimit=0, HighLimit=0
OK - VDimmP1AB, State=Enabled, Reading=1.195, LowLimit=1.062, HighLimit=1.587
OK - VDimmP1DE, State=Enabled, Reading=1.188, LowLimit=1.062, HighLimit=1.587
OK - VDimmP2AB, State=Absent, Reading=0, LowLimit=0, HighLimit=0
OK - VDimmP2DE, State=Absent, Reading=0, LowLimit=0, HighLimit=0
OK - 5VSB, State=Enabled, Reading=4.93, LowLimit=4.27, HighLimit=5.53
OK - 3.3VSB, State=Enabled, Reading=3.282, LowLimit=2.823, HighLimit=3.656
OK - 1.8V PCH, State=Enabled, Reading=1.743, LowLimit=1.608, HighLimit=1.995
OK - PVNN PCH, State=Enabled, Reading=0.976, LowLimit=0.778, HighLimit=1.108
OK - 1.05V PCH, State=Enabled, Reading=1.034, LowLimit=0.894, HighLimit=1.342
OK - 12VSB, State=Enabled, Reading=11.856, LowLimit=10.296, HighLimit=13.236
```

4.1.5 Getting Health Information on the Power Supplies

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> check -t ps
```

Example

```
$ ./nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN check -t ps
```

Console Output:

```
OK - Power Supply Bay 1, State=Enabled, Model=PWS-1K62A-1R
CRITICAL - Power Supply Bay 2, State=Enabled
N/A - Power Supply Bay 3, State=Absent
N/A - Power Supply Bay 4, State=Absent
```

4.1.6 Getting Health Information on the Storage Component

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> check -t storage
```

Example:

```
$ ./nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN check -t storage
```

Console Output:

```
OK - SATAEmbedded:Embedded SATA Storage System, State=Enabled
OK - SATAEmbedded.0:System SATA, State=Enabled, Manufacturer=None, Model=None,
Enclosures=StorageBackplane
OK - SATAEmbedded#Volume0:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=500 MB,
  Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume1:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=115.71 GB,
  Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume2:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=1014 MB,
  Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume3:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=115.69 GB,
  Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume4:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=115.69 GB,
  Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume5:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=49.98 GB,
  Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume6:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=7.69 GB,
  Drives=[StorageBackplane#Disk.Bay.3]
OK - SATAEmbedded#Volume7:SATA Bay 3, State=Enabled, Type=RawDevice, Capacity=57.96 GB,
  Drives=[StorageBackplane#Disk.Bay.3]
OK - StorageBackplane#Disk.Bay.3, State=Unknown, S/N=9XE0HT8W
```

4.1.7 Getting Performance Data

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> check -t perf
```

Example:

```
$ ./nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN check -t perf
```

Console Output:

```
Power Consumption=78
OK - FAN8, State=Enabled, Reading=900RPM, LowLimit=500, HighLimit=25400
OK - FAN9, State=Enabled, Reading=1700RPM, LowLimit=500, HighLimit=25400
OK - CPU1 Temp, State=Enabled, Reading=53C, LowLimit=0, HighLimit=102
OK - PCH Temp, State=Enabled, Reading=36C, LowLimit=0, HighLimit=85
OK - System Temp, State=Enabled, Reading=28C, LowLimit=0, HighLimit=85
OK - Peripheral Temp, State=Enabled, Reading=33C, LowLimit=0, HighLimit=85
OK - MB 10G Temp, State=Enabled, Reading=34C, LowLimit=0, HighLimit=100
OK - VRMCpu1IN Temp, State=Enabled, Reading=37C, LowLimit=0, HighLimit=100
OK - VRMCpu1IO Temp, State=Enabled, Reading=39C, LowLimit=0, HighLimit=100
OK - VRMCpu2IN Temp, State=Enabled, Reading=29C, LowLimit=0, HighLimit=100
OK - VRMCpu2IO Temp, State=Enabled, Reading=28C, LowLimit=0, HighLimit=100
OK - VRMP1AB Temp, State=Enabled, Reading=36C, LowLimit=0, HighLimit=100
OK - VRMP1DE Temp, State=Enabled, Reading=34C, LowLimit=0, HighLimit=100
OK - VRMP2AB Temp, State=Enabled, Reading=31C, LowLimit=0, HighLimit=100
OK - VRMP2DE Temp, State=Enabled, Reading=29C, LowLimit=0, HighLimit=100
OK - P1-DIMMA1 Temp, State=Enabled, Reading=33C, LowLimit=0, HighLimit=85
```

4.2 Hardware Information

The hardware information command is used to get the hardware and firmware information. The detailed information of each type of component or sensor is summarized in the table below.

Component	Attributes
Computer System	<ul style="list-style-type: none">- Model- Total Cores- Total Memory (GB)- UUID- Serial Number- SKU- BIOS Version
Processor	<ul style="list-style-type: none">- Manufacturer- Model- Socket- Total Cores- Total Threads- Max Speed (MHz)
Memory	<ul style="list-style-type: none">- Manufacturer- Part Number- Serial Number- Capacity- Type- Speed (MHz)- Location
Network	<ul style="list-style-type: none">- Name- Description- MAC Address- IPv4<ul style="list-style-type: none">• Address• Subnet Mask- IPv6<ul style="list-style-type: none">• Address
Storage	<ul style="list-style-type: none">- Controllers<ul style="list-style-type: none">• Name• Description• Manufacturer• Serial Number• Firmware Version- Drives<ul style="list-style-type: none">• Name• Manufacturer• Model• Serial Number- Volumes

	<ul style="list-style-type: none"> • Type • Capacity
Chassis	<ul style="list-style-type: none"> - Manufacturer - Serial Number - Asset Tag - Type
Power Supply	<ul style="list-style-type: none"> - Name - State - Type - Model - Serial Number - Firmware Version - Input Voltage - Power Output
Manager (BMC)	<ul style="list-style-type: none"> - Model - Firmware Version - MAC Address - IPv4 Address - IPv6 Address
BIOS	<ul style="list-style-type: none"> - Firmware Version

4.2.1 Getting Hardware Information

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> hwinfo [-f <file>]
```

optional arguments:

```
-f <file>, --file <file>    file location for exporting hardware information (JSON)
```

Example:

```
$ ./nagios_supermicro_redfish -i10.136.33.151 -uADMIN -pADMIN hwinfo
```

Console Output:

```
get system information...done
get processor information...done
get memory information...done
get network information...done
get chassis information...done
get power supply information...done
get storage information...done
get ipmi information...done
get bios information...done

Hardware Information
-----
{
  "Computer System": {
    "Model": "Super Server",
    "Total Cores": 20,
    "Total Memory (GB)": 64,
    "UUID": "00000000-0000-0000-0000-0CC47AFBAD66",
    "Serial Number": " ",
    "SKU": "To be filled by O.E.M.",
    "BIOS version": "2.1"
  }
}
```

```

},
"Processor": [
  {
    "Manufacturer": "Intel(R) Corporation",
    "Model": "Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz",
    "Socket": "CPU1",
    "Total Cores": 10,
    "Total Threads": 20,
    "Max Speed (MHz)": 4500
  },
  {
    "Manufacturer": "Intel(R) Corporation",
    "Model": "Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz",
    "Socket": "CPU2",
    "Total Cores": 10,
    "Total Threads": 20,
    "Max Speed (MHz)": 4500
  }
],
"Memory": [
  {
    "Manufacturer": "Micron Technology",
    "Part Number": "36ASF2G72PZ-2G1A2",
    "Serial Number": "0CC33709",
    "Capacity": "16 GB",
    "Type": null,
    "Speed (MHz)": [
      2133
    ],
    "Location": "P2-DIMMA2"
  },
  {
    "Manufacturer": "Micron Technology",
    "Part Number": "36ASF2G72PZ-2G1A2",
    "Serial Number": "0CC33720",
    "Capacity": "16 GB",
    "Type": null,
    "Speed (MHz)": [
      2133
    ],
    "Location": "P2-DIMMA1"
  },
  {
    "Manufacturer": "Micron Technology",
    "Part Number": "36ASF2G72PZ-2G1A2",
    "Serial Number": "0CC23D33",
    "Capacity": "16 GB",
    "Type": null,
    "Speed (MHz)": [
      2133
    ],
    "Location": "P1-DIMMA2"
  },
  {
    "Manufacturer": "Micron Technology",
    "Part Number": "36ASF2G72PZ-2G1A2",
    "Serial Number": "0CC23D2A",
    "Capacity": "16 GB",
    "Type": null,
    "Speed (MHz)": [
      2133
    ],
    "Location": "P1-DIMMA1"
  }
],
"Network": [
  {
    "Name": "enp94s0f0",
    "Description": "Intel Corporation I350 Gigabit Network Connection",
    "MAC Address": "0c:c4:7a:fb:ad:66",
    "IPv4": [

```

```

        {
            "Address": "N/A",
            "Subnet Mask": "N/A"
        }
    ],
    "IPv6": [
        {
            "Address": "N/A"
        }
    ]
},
{
    "Name": "enp94s0f1",
    "Description": "Intel Corporation I350 Gigabit Network Connection",
    "MAC Address": "0c:c4:7a:fb:ad:67",
    "IPv4": [
        {
            "Address": "10.163.55.62",
            "Subnet Mask": "255.255.240.0"
        }
    ],
    "IPv6": [
        {
            "Address": "fe80::522e:59bc:b0c9:3ba5"
        }
    ]
},
{
    "Name": "enp94s0f0",
    "Description": "Intel Corporation I350 Gigabit Network Connection",
    "MAC Address": "0c:c4:7a:fb:ad:66",
    "IPv4": [
        {
            "Address": "N/A",
            "Subnet Mask": "N/A"
        }
    ],
    "IPv6": [
        {
            "Address": "N/A"
        }
    ]
},
{
    "Name": "enp94s0f1",
    "Description": "Intel Corporation I350 Gigabit Network Connection",
    "MAC Address": "0c:c4:7a:fb:ad:67",
    "IPv4": [
        {
            "Address": "10.163.55.62",
            "Subnet Mask": "255.255.240.0"
        }
    ],
    "IPv6": [
        {
            "Address": "fe80::522e:59bc:b0c9:3ba5"
        }
    ]
},
{
    "Name": "virbr0",
    "Description": "N/A",
    "MAC Address": "52:54:00:0a:3d:db",
    "IPv4": [
        {
            "Address": "192.168.122.1",
            "Subnet Mask": "255.255.255.0"
        }
    ],
    "IPv6": [
        {

```

```

        "Address": "N/A"
    }
]
},
"Chassis": {
    "Manufacturer": "Supermicro",
    "Serial Number": "",
    "Asset Tag": "",
    "Type": "RackMount"
},
"Power Supply": [
    {
        "Name": "Power Supply Bay 1",
        "State": "Enabled",
        "Type": "AC",
        "Model": "PWS-1K23A-1R",
        "Serial Number": "P1K2BCYWWA00001",
        "Firmware Version": "REV1.0",
        "ACLowLine Input Voltage (V)": 116,
        "Power Output (W)": 52
    },
    {
        "Name": "Power Supply Bay 2",
        "State": "Enabled",
        "Type": "AC",
        "Model": "PWS-1K23A-1R",
        "Serial Number": "P1K2ACF19GB0138",
        "Firmware Version": "REV1.0",
        "ACLowLine Input Voltage (V)": 117,
        "Power Output (W)": 73
    },
    {
        "Name": "Power Supply Bay 3",
        "State": "Absent"
    },
    {
        "Name": "Power Supply Bay 4",
        "State": "Absent"
    }
],
"Storage": [
    {
        "Id": "HA-RAID",
        "Controllers": [
            {
                "Id": "0",
                "Name": "External RAID",
                "Description": "External RAID (PCIe card: Onboard, slot: 3)",
                "Manufacturer": "AVAGO",
                "Model": "3108",
                "Serial Number": "FW-AL85PGVAARBWA",
                "Firmware Version": "4.680.00-8290"
            }
        ]
    },
    {
        "Id": "0",
        "Name": "BCM RAID Controller 0 Virtual Disk 0",
        "Volume Type": "Mirrored",
        "Block Size": 512,
        "Capacity": "744.69 GB",
        "Encrypted": false,
        "Drives": [
            {
                "Drive": "Disk.Bay.0",
                "Enclosure": "HA-RAID.0.StorageEnclosure.0"
            },
            {
                "Drive": "Disk.Bay.1",
                "Enclosure": "HA-RAID.0.StorageEnclosure.0"
            }
        ]
    }
]

```



```

    }
  ]
}
],
"Enclosures": [
{
  "Id": "HA-RAID.0.StorageEnclosure.0",
  "Drives": [
    {
      "Name": "Disk.Bay.0",
      "Manufacturer": "SEAGATE",
      "Model": "ST800FM0053",
      "Serial Number": "Z3G01KD8"
    },
    {
      "Name": "Disk.Bay.1",
      "Manufacturer": "SEAGATE",
      "Model": "ST800FM0053",
      "Serial Number": "Z3G01KGM"
    }
  ]
}
]
},
{
  "Id": "SATAEmbedded",
  "Controllers": [
    {
      "Id": "0",
      "Name": "System SATA",
      "Description": "System SATA",
      "Manufacturer": null,
      "Model": null,
      "Serial Number": null,
      "Firmware Version": null
    }
  ],
  "Volumes": [
    {
      "Id": "0",
      "Name": "SATA Bay 2",
      "Volume Type": "RawDevice",
      "Block Size": null,
      "Capacity": "1014 MB",
      "Encrypted": false,
      "Drives": [
        {
          "Drive": "Disk.Bay.2",
          "Enclosure": "StorageBackplane"
        }
      ]
    }
  ],
  {
    "Id": "1",
    "Name": "SATA Bay 2",
    "Volume Type": "RawDevice",
    "Block Size": null,
    "Capacity": "2.98 GB",
    "Encrypted": false,
    "Drives": [
      {
        "Drive": "Disk.Bay.2",
        "Enclosure": "StorageBackplane"
      }
    ]
  },
  {
    "Id": "2",
    "Name": "SATA Bay 2",
    "Volume Type": "RawDevice",
    "Block Size": null,

```

```

        "Capacity": "25.82 GB",
        "Encrypted": false,
        "Drives": [
            {
                "Drive": "Disk.Bay.2",
                "Enclosure": "StorageBackplane"
            }
        ]
    },
    "Enclosures": [
        {
            "Id": "StorageBackplane",
            "Drives": [
                {
                    "Name": "Disk.Bay.2",
                    "Manufacturer": null,
                    "Model": null,
                    "Serial Number": "B4500757042400103641"
                }
            ]
        }
    ]
},
"BMC": {
    "Model": "ASPEED",
    "Firmware Version": "1.55",
    "MAC Address": "0C:C4:7A:88:21:27",
    "IPv4 Address": [
        "10.163.55.148"
    ],
    "IPv6 Address": [
        "fe80::ec4:7aff:fe88:2127"
    ]
},
"BIOS": {
    "Firmware Version": "BIOS Date: 07/06/2018 Rev 2.1"
}
}

```

Example: Exporting to a File in JSON Format

```

$ ./nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN hwinfo -f
hwinfo.json

```

4.3 BIOS Firmware Management

The BIOS firmware management commands are used to manage BIOS configurations and firmware. The BIOS configuration management only supports the Purley and the platforms of later versions support Human Interface Infrastructure (HII).

4.3.1 Listing BIOS Configurations

Use the command to list the current BIOS configurations.

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> bios config list
```

Example:

```
$ ./nagios_supermicro_redfish -i10.136.160.176 -uADMIN -pADMIN bios config list
```

Console Output:

Item	Value
-----	-----
ADDDCSparing	Disable
AES-NI	Enable
ATS	Enable
Above4GDecoding	Enabled
AdjacentCachePrefetch	Enable
AggressivelinkPowerManagement	Disable
AggressivelinkPowerManagement\$2	Disable
AutonomousCoreC-State	Disable
Bitspersecond\$2	115200
BootOption#1\$3	Hard Disk: ST9250610NS
BootOption#1\$4	UEFI: Built-in EFI Shell
BootOption#1\$5	P1: ST9250610NS
BootOption#1\$6	IBA XE Slot 0100 v2205
BootOption#10	UEFI CD/DVD
BootOption#11	UEFI USB Hard Disk
BootOption#12	UEFI USB CD/DVD
BootOption#13	UEFI USB Key
BootOption#14	UEFI USB Floppy
BootOption#15	UEFI USB Lan
BootOption#16	UEFI Network
BootOption#17	UEFI AP:UEFI: Built-in EFI Shell
BootOption#2\$3	CD/DVD
BootOption#2\$4	IBA XE Slot 0101 v2205
BootOption#3\$3	USB Hard Disk
BootOption#4\$3	USB CD/DVD
BootOption#5\$3	USB Key
BootOption#6\$3	USB Floppy
BootOption#7\$3	USB Lan
BootOption#8\$3	Network:IBA XE Slot 0100 v2205
BootOption#9\$2	UEFI Hard Disk
Bootmodeselect	DUAL
BootupNumLockState	On
CPUC6report	Auto
ChangeSettings	Auto
ChangeSettings\$2	Auto
CoherencySupport(Non-Isoch)	Enable
ConfigureSATAas	AHCI
ConfiguresATAas	AHCI
ConsoleRedirection	0
ConsoleRedirection\$2	1

4.3.2 Exporting BIOS Configurations

Use the command to export a BIOS configuration file in JSON format. User can modify the value of the current configuration directly and update the BIOS configuration via the steps in [4.3.3 Update BIOS Configuration](#). For the format of the file, see [3.2 Format of BIOS configuration JSON File](#)

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> bios config export -f <file>
```

optional arguments:

```
-f <file>, --file <file>    file location for exporting configuration (JSON)
```

Example:

```
$ ./nagios_supermicro_redfish -i10.136.160.176 -uADMIN -pADMIN bios config export  
-f bios_config.json
```

4.3.3 Updating BIOS Configurations

The step for updating BIOS configuration:

1. Get the BIOS configuration file by running the command mentioned in [4.3.2 Export BIOS Configuration](#)
2. Edit the **Value** in the attribute that needs to be updated. The values are listed in the key, **AllowableValue**.
3. Run the command **bios config update** to update the configurations.
4. Reboot the system for the new configuration to be applied.

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> bios update -f <file>
```

optional arguments:

```
-f <file>, --file <file>    configuration file (JSON)
```

Example:

```
./nagios_supermicro_redfish -i10.136.160.176 -uADMIN -pADMIN bios config update  
-f bios_config.json
```

Console Output:

```
- check attributes
Name                | Current                | Update
-----
OptionROMMessages   | Keep Current           | Force BIOS
PowerButtonFunction | 4 Seconds Override     | Instant Off

- update attributes

- BIOS Configuration Pending Settings
  These settings will be applied on next system reboot
Item                | Value
-----
OptionROMMessages   | Force BIOS
PowerButtonFunction | Instant Off
```

4.3.4 Listing BIOS Firmware Information

Use this command to get information on the BIOS firmware.

The information contains:

Field	Description
Name	The name of the firmware type
Version	Firmware version

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> bios fw info
```

Example:

```
./nagios_supermicro_redfish -i10.136.33.92 -uADMIN -pADMIN bios fw info
```

Console Output:

```
Name          | SUPERMICRO BMC BIOS
Version       | BIOS Date: 07/05/2018 Rev 2.1
```

4.3.5 Updating BIOS Firmware

This is the process of how the BMC firmware is updated.

1. The firmware file is checked to see if it exists
2. Update mode is entered
3. An image file is uploaded; this will take a few minutes.
4. Firmware is now being updated
5. During the update, a message like **Updating** or **ReadyToUpdate** is shown.

After the update is finished, the system must be rebooted to apply the new firmware.



Notes:

- If an error occurs during the update process, the firmware update will be cancelled and log firmware out of the update mode.
 - If the firmware is already in the update mode, it might be due to previously un-completed process. The command will cause an error and log firmware out of the update mode. You need to run the command again to update the firmware.
 - **--preserve_nv** option is used to preserve the BIOS NVRAM data. Unless you are familiar with BIOS NVRAM, do not use this option.
 - **--preserve_mer** option is used to preserve the ME firmware region. Unless you are familiar with ME firmware region, do not use this option.
-

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password>
                           bios fw update -f <file>
```

optional arguments:

```
-f <file>, --file <file>    firmware image
--preserve_nv                reserve NVRAM data
--preserve_mer               reserve ME firmware region
```

Example:

```
./nagios_supermicro_redfish -i10.136.33.92 -uADMIN -pADMIN bios fw update -f  
~/firmware/x10sled5.C17
```

Console Output:

```
1. Check if an image exists.  
2. It is now entering the update mode.  
3. Uploading image... Please wait, this will take a while!  
[>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>] 100.0%  
4. Start updating, please do not power off the system.  
.....  
.....  
.....  
Update is successful. Please reboot your system.
```

4.4 BMC Firmware Management

4.4.1 Listing BMC Firmware Information

Use this command to get the information on the BMC firmware.

The information contains:

Field	Description
Name	The name of the firmware type
Version	Firmware version

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> bmc fw info
```

Example:

```
./nagios_supermicro_redfish -i10.136.33.92 -uADMIN -pADMIN bmc fw info
```

Console Output:

```
Name          | SUPERMICRO BMC Firmware
Version       | 01.55
```

4.4.2 Updating BMC Firmware

This is the process of how BMC firmware is updated.

1. The firmware file is checked to see if it exists.
2. Update mode is entered.
3. An image file is uploaded; this will take a few minutes.
4. Firmware is now being updated.
5. The progress is shown in percentage terms.
6. If the status indicates **100%**, the firmware update is finished.
7. After the update is finished, the BMC will reset, which will take a few minutes.



Notes:

- If an error occurs during the update process, the firmware update will be cancelled and log firmware out of the update mode.
 - If the firmware is already in the update mode, it might be due to previously un-completed process. The command will cause an error and log firmware out of the update mode. You need to run the command again to update the firmware.
-

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password>
```

optional arguments:

```
-f <file>, --file <file>      firmware image
--overwrite_cfg                restore the BMC's factory default setting
--overwrite_sdr                restore the SDR defaults
--overwrite_sslcert            restore the default SSL certificate
```

Example:

```
./nagios_supermicro_redfish -i10.136.33.92 -uADMIN -pADMIN bmc fw update -f  
~/firmware/REDFISH_X10_369.bin
```

Console Ooutput:

```

1. Check if an image exists.
2. It is now entering the update mode.
3. Uploading image... Please wait, this will take a while!
[>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>] 100.0%
4. Start updating, please do not power off the system.
[>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>] 100.0%
Update is successful.
Please wait for BMC to complete system reboot.
```

4.4.3 BMC Cold Reboot

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> bmc reset
```

4.5 Event Service

The Redfish event service provides a mechanism for client to create subscriptions to receive events. The client needs to provide the URI of the event receiver and the event types to be sent. If the event is triggered, the event will be send to the event receiver. The event-service command set provides the interface to manage the subscription. For more information about the event receiver, see [5. Event Receiver](#)

The event types include **StatusChange**, **ResourceUpdated**, **ResourceAdded**, **ResourceRemoved** and **Alert**.

4.5.1 Listing Subscriptions

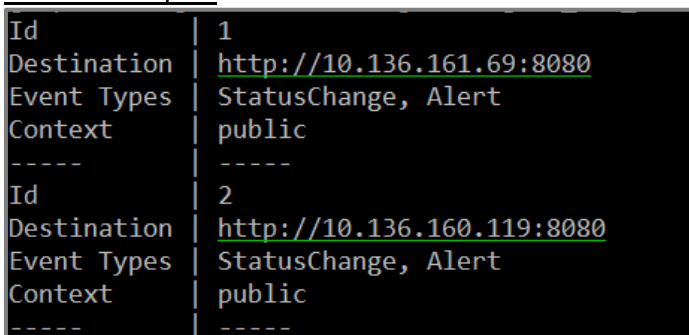
Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> event-service list
```

Example:

```
$ ./nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN event-service list
```

Console Output:



```
Id          | 1
Destination | http://10.136.161.69:8080
Event Types | StatusChange, Alert
Context     | public
-----
Id          | 2
Destination | http://10.136.160.119:8080
Event Types | StatusChange, Alert
Context     | public
-----
```

4.5.2 Adding a Subscription

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password>
                             event-service add <dest>
                             [--type [EVENT_TYPE]]
                             [--context [CONTEXT]]
```

positional arguments:

<dest> URL of destination of destination(<ip>:<port>)

optional arguments:

--type [EVENT_TYPE] event type: StatusChange, ResourceUpdated, ResourceAdded, ResourceRemoved and Alert (default: StatusChange, Alert)

--context [CONTEXT] context (default: public)

Example:

```
./nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN event-service add  
10.136.161.69:8080 -type StatusChange ResourceUpdated ResourceAdded  
ResourceRemoved Alert --context private
```

4.5.3 Deleting a Subscription

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> event-service del <id>
```

positional arguments:

```
<id>                subscription id
```

Example:

```
$ ./nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN event-service del 3
```

5. Event Receiver

The Redfish server will POST the event to the event receiver that has subscribed to event service. The Supermicro event receiver is a simple Web server, and it receives events and writes the event to a log file by a remote IP.

5.1 Supermicro Event Receiver

Usage:

```
supermicro_event_receiver [-h] -i <ip> [-p <port>] [--log_dir <log directory>]
```

optional arguments:

-h, --help	show this help message and exit
-i <ip>	event receiver IP
-p <port>	event receiver port
--log_dir <log directory>	event log directory

Example:

```
$ ./supermicro_event_receiver -i 10.136.160.119 -p 8080
```

Console Output:

```
[supermicro@localhost bin]$ ./supermicro_event_receiver -i10.136.161.101 -p8081
smc_event_receiver 1.0.0 build 180503
Log directory: /home/supermicro/py-workspace/test/nagios_supermicro_redfish/bin/log
* Running on http://10.136.161.101:8081/ (Press CTRL+C to quit)
```

5.2 Event Log

The event will write to a log file. The format of the filename is
event_<remote ip>.log

The format of an event log:

```
<severity> - <remote ip>, #<event id> <event type>, Message[<message id>]=<message>
```

Contacting Supermicro

Headquarters

Address: Super Micro Computer, Inc.
980 Rock Ave.
San Jose, CA 95131 U.S.A.
Tel: +1 (408) 503-8000
Fax: +1 (408) 503-8008
Email: marketing@supermicro.com (General Information)
support@supermicro.com (Technical Support)
Web Site: www.supermicro.com

Europe

Address: Super Micro Computer B.V.
Het Sterrenbeeld 28, 5215 ML
's-Hertogenbosch, The Netherlands
Tel: +31 (0) 73-6400390
Fax: +31 (0) 73-6416525
Email: sales@supermicro.nl (General Information)
support@supermicro.nl (Technical Support)
rma@supermicro.nl (Customer Support)

Asia-Pacific

Address: Super Micro Computer, Inc.
3F, No. 150, Jian 1st Rd.
Zhonghe Dist., New Taipei City 235
Taiwan (R.O.C)
Tel: +886-(2) 8226-3990
Fax: +886-(2) 8226-3992
Web Site: www.supermicro.com.tw
Technical Support:
Email: support@supermicro.com.tw
Tel: +886-(2)-8226-3990