



Supermicro Redfish Plug-in for Nagios Core

User's Guide

Revision 1.3

The information in this USER'S GUIDE 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.

DISCLAIMER OF WARRANTY ON SOFTWARE AND MATERIALS. You expressly acknowledge and agree that use of the Software and Materials is at your sole risk. FURTHERMORE, SUPER MICRO COMPUTER INC. DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE OR MATERIALS IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY SUPER MICRO COMPUTER INC. OR SUPER MICRO COMPUTER INC. AUTHORIZED REPRESENTATIVE SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY. SHOULD THE SOFTWARE AND/OR MATERIALS PROVE DEFECTIVE, YOU (AND NOT SUPER MICRO COMPUTER INC. OR A SUPER MICRO COMPUTER INC. AUTHORIZED REPRESENTATIVE) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICE, REPAIR, OR CORRECTION.

LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES INCLUDING NEGLIGENCE, SHALL SUPER MICRO COMPUTER INC. BE LIABLE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES THAT RESULT FROM THE USE OR INABILITY TO USE THE SOFTWARE OR MATERIALS, EVEN IF SUPER MICRO COMPUTER INC. OR A SUPER MICRO COMPUTER INC. AUTHORIZED REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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.

Manual Revision 1.3

Release Date: 2024/06/26

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 © 2018-2024 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 document.
2020/09/26	1.1	Added support for updating BMC/BIOS on X12/H12 platforms. Added get log service command.
2023/11/28	1.2	Added the get redfish version command. Added the firmware inventory command. Added the crashdump command. Added PCIe devices to the hwinfo command. Expanded the check command to include additional component types: CPU, memories, NIC, and GPU. Added support for updating BIOS on Tatlow and later platforms.
2024/06/26	1.3	Added support for X14/H14 platforms. Added support for the HTTPS event receiver Added the power control command.

Contents

1. Overview	7
1.1 Prerequisites	8
1.1.1 System Requirements.....	8
1.1.2 Firmware Requirements	8
1.1.3 License Requirements.....	8
1.1.4 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	15
4. Managing Server	17
4.1 Health Information	17
4.1.1 Getting Health Information on the Entire System.....	19
4.1.2 Getting Health Information on the Fan Sensors.....	21
4.1.3 Getting Health Information on the Temperature Sensors	22
4.1.4 Getting Health Information on the Voltage Sensors	23
4.1.5 Getting Health Information on the Power Supplies	23
4.1.6 Getting Health Information on the Storage Component	24
4.1.7 Getting Health Information on the CPU Component	24
4.1.8 Getting Health Information on the Memory Component	25
4.1.9 Getting Health Information on the NIC Component	25
4.1.10 Getting Health Information on the GPU Component.....	25
4.1.11 Getting Performance Data.....	26
4.1.12 Filtering Health Status by Severity.....	26
4.2 Hardware Information	27
4.2.1 Getting Hardware Information	28
4.3 BIOS Firmware Management.....	37
4.3.1 Listing BIOS Configurations.....	37

4.3.2	Exporting BIOS Configurations	37
4.3.3	Updating BIOS Configurations	38
4.3.4	Listing BIOS Firmware Information.....	39
4.3.5	Updating BIOS Firmware	39
4.4	BMC Firmware Management	41
4.4.1	Listing BMC Firmware Information.....	41
4.4.2	Updating BMC Firmware	41
4.4.3	BMC Cold Reboot.....	42
4.5	Event Service.....	43
4.5.1	Listing Subscriptions	43
4.5.2	Adding a Subscription	43
4.5.3	Deleting a Subscription.....	44
4.6	Log Service	45
4.6.1	Getting Health Event Log Information.....	45
4.6.2	Getting Information of Maintenance Event Log.....	45
4.6.3	Getting Health Event Logs	46
4.6.4	Getting Maintenance Event Logs.....	46
4.6.5	Limiting the Number of Output Event Logs.....	47
4.7	Firmware Inventory Management.....	48
4.7.1	Listing Firmware Inventory Information.....	48
4.7.2	Updating BMC Firmware Golden Image.....	48
4.7.3	Recovering BMC Firmware	49
4.7.4	Updating BIOS Firmware Golden Image	49
4.7.5	Recovering BIOS Firmware	49
4.8	Crashdump Service	50
4.8.1	Generating a Crashdump Log	50
4.8.2	Downloading a Crashdump Log.....	50
4.9	Redfish Version	51
4.9.1	Getting the Redfish Version.....	51
4.10	Power control	51
4.10.1	Power status	51

4.10.2	Power on.....	51
4.10.3	Power down immediately.....	51
4.10.4	Graceful shutdown	52
4.10.5	Power cycle.....	52
4.10.6	Power reset.....	52
4.10.7	AC Power cycle	52
4.11	BMC Redfish Raw Data	53
4.11.1	Get BMC redfish API Root.....	53
4.11.2	Get BMC redfish API	53
5.	Event Receiver	55
5.1	Supermicro Event Receiver.....	55
5.2	Event Log.....	56
Appendix A Third-Party Software		57
Contacting Supermicro		58

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 BIOS and BMC firmware
- Updating BIOS configurations
- Hardware and firmware inventory
- Managing event subscriptions
- System event and maintenance event logs
- Crash dump log
- Power control

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

Firmware Image	Requirements
BMC	X10 ATEN platform (REDFISH_X10): 3.77 or later. X11 ATEN platform (SMT_X11): 1.55 or later. X12 ATEN platform (SMT_X12): 1.00 or later H12 ATEN platform (SMT_H12): 1.00 or later X13 ATEN platform (SMT_X13): 1.00 or later H13 ATEN platform (SMT_H13): 1.00 or later X14 ATEN platform (SMT_X14): 1.00 or later H14 ATEN platform (SMT_H14): 1.00 or later

1.1.3 License Requirements

SFT-DCMS-Single per-node license is required.



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

1.1.4 Additional Package Requirements for Remote Servers

Features	Requirements
<ul style="list-style-type: none">PCH SATA informationIntel RSTe informationEthernet interface information	TAS_1.6.0_build.200601 or later

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. --show_all (Optional) Show all components	Gets the health information of systems and storage systems. It shows the failure components if the system is in abnormal status by default.

	-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 cpu cpu components memory memory components nic nic components gpu gpu components perf fan, temp sensors and power consumption	
	-s, --filter_severity <severity> (Optional) Filter components by specific severity <severity> ok component with ok status warning component with warning status critical component with critical status	
Hardware Information		
Command	Option	Description
hwinfo	-f <file>, --file <file> (Optional) Export to file.	Gets the hardware information.
BIOS Management		
Command	Option	Description
bios config list		Lists all BIOS configurations.
bios config export	-f <file>, --file <file> File location.	Exports the current BIOS configurations to an assigned file.
bios config update	-f <file>, --file <file> File location.	Updates the BIOS configurations with the given configuration file.
bios fw info		Gets 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. --backup (Optional, ROT only) Back up the active image. --overwrite_oa (Option, Tatlow and later platform only) Overwrite OA --overwrite_setup_conf (Option, Tatlow and later platform only) Overwrite BIOS Setup Configuration --overwrite_setup_pwd (Option, Tatlow and later platform only) Overwrite BIOS Setup password --overwrite_secure_bootkeys (Option, Tatlow and later platform only) Overwrite BIOS Secure Boot Keys --overwrite_boot_conf (Option, Tatlow and later platform only) Overwrite BIOS Boot Options Configuration	Updates the BIOS firmware with the given image file.
BMC Management		
Command	Option	Description
bmc fw info		Gets 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. --backup (Optional, ROT only) Back up the active image.	Updates the BMC firmware with the given image file.
bmc reset		BMC cold reboot
Event Service		
Command	Option	Description
event-service list		Lists 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") --id <id> The --id value ranges from 1 to 16. Note the option is only available on X12 and later platforms. --status [STATUS_TYPE] status type: enabled, disabled (default: enabled) Note the option is only available on X12 and later platforms.	Adds a subscription.
event-service del	id Subscription ID	Deletes a subscription ID.
Log Service		
Command	Option	Description
log info	-t, --type <type> (Optional) Show specific type of components <type> health (default) health event log syslog maintenance event log	Gets log information.
log list	-t, --type <type> (Optional) Show specific type of components <type> health (default) health event log syslog maintenance event log	Shows the health (by default) or maintenance event log by the specific type.
	-l, --last <count> (Optional) Show count number of logs <count> get the last count log entries	

Firmware Inventory Management		
Command	Command	Description
fw_inventory info		Get firmware inventory information.
Fw_inventory update	bmc, bios Action target --recover Recover BMC or BIOS from backup image --Update_golden Update the current BMC or BIOS to golden image	Updates the current active as golden template and recover BMC or BIOS from the backup image.
Crashdump service		
Command	Command	Description
crashdump generate		Generate the crash dump log.
crashdump download	-f <file>, --file <file> File location.	Download the crash dump log and save as tar.gz file
Redfish version		
Command	Command	Description
redfish_ver		Show the redfish version.
Command	Command	Description
power	-t, --type <type> Power action <type> status Show current power status on Power on force_off Power down immediately graceful_shutdown Graceful shutdown cycle Power cycle reset Power reset ac_cycle AC Power cycle	Power control command.
Command	Command	Description
raw	-api	Retrieve the full response from the BMC Redfish API

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 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
CPU	<ul style="list-style-type: none">• health• state• model• speed• tdp watts• core• thread
Memory	<ul style="list-style-type: none">• health• state• manufacturer• part number• serial number• capacity
NIC(AOC)	<ul style="list-style-type: none">• health• state• model• serial number• PCIe slot
GPU	<ul style="list-style-type: none">• health• state• model• serial number• firmware version
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 • mode
Component	Attributes
Storage	Storage System <ul style="list-style-type: none"> • health • state Controller <ul style="list-style-type: none"> • health • state • controller ID • 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 plugin return code are defined by Nagios: **OK**, **WARNING**, **CRITICAL**, and **UNKNOWN**. The Supermicro Redfish Plug-in defines several types of health status, including OK, WARNING, CRITICAL, UNKNOWN, ACTIVE, ACTIVE SYNC, and FAIL. FAIL is considered a CRITICAL return code. ACTIVE and ACTIVE SYNC are health statuses for the RAID system and are considered OK return codes. The **state** is defined in Redfish which indicates the known state of the resource.

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

Use the `check` command to get the overall status of the computer system and storage systems. The statuses of storage 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: Showing 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]e
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 Health Information on the CPU Component

Usage:

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

Example:

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

Console Output:

```
OK - CPU1, State=Enabled, Model=AMD EPYC 9384X 32-Core Processor, Speed=3100
MHz, Core=32, Threads=64
OK - CPU2, State=Enabled, Model=AMD EPYC 9384X 32-Core Processor, Speed=3100
MHz, Core=32, Threads=64
```

4.1.8 Getting Health Information on the Memory Component

Usage:

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

Example:

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

Console Output:

```
OK      - P1-DIMMA1, State=Enabled, Manufacturer=Samsung,  
P/N=M321R4GA3BB6-CQKEG, S/N=80CE01221102AA61AB, Capacity=32 GiB  
OK      - P1-DIMMB1, State=Enabled, Manufacturer=Samsung,  
P/N=M321R4GA3BB6-CQKEG, S/N=80CE01221102AA63C5, Capacity=32 GiB  
OK      - P1-DIMMC1, State=Enabled, Manufacturer=Samsung,  
P/N=M321R4GA3BB6-CQKEG, S/N=80CE01221102AA6148, Capacity=32 GiB  
OK      - P1-DIMMD1, State=Enabled, Manufacturer=Samsung,  
P/N=M321R4GA3BB6-CQKEG, S/N=80CE01221102AA660D, Capacity=32 GiB  
OK      - P1-DIMME1, State=Enabled, Manufacturer=Samsung,  
P/N=M321R4GA3BB6-CQKEG, S/N=80CE01221102AA6623, Capacity=32 GiB  
OK      - P1-DIMMF1, State=Enabled, Manufacturer=Samsung,  
P/N=M321R4GA3BB6-CQKEG, S/N=80CE01221102AA62E9, Capacity=32 GiB  
OK      - P1-DIMMG1, State=Enabled, Manufacturer=Samsung,  
P/N=M321R4GA3BB6-CQKEG, S/N=80CE01221102AA6475, Capacity=32 GiB
```

4.1.9 Getting Health Information on the NIC Component

This only shows AOC NIC information.

Usage:

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

Example:

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

Console Output:

```
OK      - NIC1, State=Enabled, Model=MCX75310AAS-HEAT, PCIe Slot=System Slot 5,  
S/N=MT2236X03968  
OK      - NIC2, State=Enabled, Model=MCX75310AAS-NEAT, PCIe Slot=System Slot 6,  
S/N=MT2309XZ0E0U  
OK      - NIC3, State=Enabled, Model=MCX653105A-ECAT, PCIe Slot=System Slot 8,  
S/N=MT2229J48859  
OK      - NIC4, State=Enabled, Model=AOC-S100G-b2C, PCIe Slot=System Slot 13,  
S/N=OA227S005976  
OK      - NIC5, State=Disabled, Model=AOM-PTG-I2T, PCIe Slot=System Slot 1,  
S/N=OA233S024151
```

4.1.10 Getting Health Information on the GPU Component

Usage:

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

Example:

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

Console Output:

```
OK      - GPU1, State=Enabled, Model=NVIDIA A100 80GB PCIe, S/N=1321322016321,
F/W version=92.00.90.00.0F
OK      - GPU2, State=Enabled, Model=NVIDIA A100 80GB PCIe, S/N=1321322016367,
F/W version=92.00.90.00.0F
OK      - GPU3, State=Enabled, Model=NVIDIA A100 80GB PCIe, S/N=1321322057488,
F/W version=92.00.90.00.0F
OK      - GPU4, State=Enabled, Model=NVIDIA A100 80GB PCIe, S/N=1321322057224,
F/W version=92.00.90.00.0F
```

4.1.11 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:

```
OK - Power Consumption=171, State=Enabled
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
```

4.1.12 Filtering Health Status by Severity

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> check --show_all
--filter_severity <severity>
```

Example:

```
$ ./nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN --show_all -
-filter_severity ok
```

Console Output:

```
OK - DIMMA1, State=Enabled, Manufacturer=SK Hynix, P/N=HMA82GR7CJR8N-WM,
S/N=42F574BA, Capacity=16 GiB
OK - DIMMD1, State=Enabled, Manufacturer=SK Hynix, P/N=HMA82GR7CJR8N-WM,
S/N=42F573E8, Capacity=16 GiB
OK - FAN, State=Enabled, Reading=1900RPM, LowLimit=500, HighLimit=25400
OK - PCH Temp, State=Enabled, Reading=55C, LowLimit=5, HighLimit=90
OK - System Temp, State=Enabled, Reading=35C, LowLimit=5, HighLimit=85
```

```

OK - Peripheral Temp, State=Enabled, Reading=41C, LowLimit=5, HighLimit=85
OK - VRMCpu Temp, State=Enabled, Reading=44C, LowLimit=5, HighLimit=100
OK - VRMAB Temp, State=Enabled, Reading=55C, LowLimit=5, HighLimit=100
OK - VRMDE Temp, State=Enabled, Reading=55C, LowLimit=5, HighLimit=100
OK - DIMMA1 Temp, State=Enabled, Reading=44C, LowLimit=5, HighLimit=85
OK - DIMMD1 Temp, State=Enabled, Reading=42C, LowLimit=5, HighLimit=85
OK - 12V, State=Enabled, Reading=12.064, LowLimit=10.272, HighLimit=13.28
OK - 5VCC, State=Enabled, Reading=5.15, LowLimit=4.28, HighLimit=5.72
OK - 3.3VCC, State=Enabled, Reading=3.265, LowLimit=2.823, HighLimit=3.775
OK - Vcpu, State=Enabled, Reading=1.808, LowLimit=1.259, HighLimit=2.087
OK - VDimmAB, State=Enabled, Reading=1.176, LowLimit=1.026, HighLimit=1.374
OK - VDimmDE, State=Enabled, Reading=1.212, LowLimit=1.026, HighLimit=1.374
OK - 5VSB, State=Enabled, Reading=5.093, LowLimit=4.343, HighLimit=5.603
OK - 3.3VSB, State=Enabled, Reading=3.235, LowLimit=2.819, HighLimit=3.763
OK - P1V8_PCH, State=Enabled, Reading=1.841, LowLimit=1.535, HighLimit=2.057
OK - PVNN_PCH, State=Enabled, Reading=1.023, LowLimit=0.897, HighLimit=1.219
OK - P1V05_PCH, State=Enabled, Reading=1.066, LowLimit=0.898, HighLimit=1.22
OK - Power Supply Bay 2, State=Enabled, Model=PWS-2K04A-1R
OK - Intrusion Sensor, State=Normal

```

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
PCIe Devices	<ul style="list-style-type: none"> • Id • State • Manufacturer • Model • Serial Number • Firmware Version

4.2.1 Getting Hardware Information

Usage:

```
nagios_supersmicro_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 pcie devices 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": "000000000-0000-0000-0000-0CC47AFBAD66",
    "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"
            }
        ]
    }
]
},
"PCIe Devices": [
    {
        "Id": "NIC10",
        "State": "Disabled",
        "Manufacturer": "Supermicro",
        "Model": "AOM-PTG-I2T",
        "Serial Number": "OA232S001695",
        "Firmware Version": "8.50 0x8000BE1E"
    }
]

```

```

    },
    {
      "Id": "GPU1",
      "State": "Enabled",
      "Manufacturer": "NVIDIA",
      "Model": "NVIDIA A16",
      "Serial Number": "1321023002322",
      "Firmware Version": "94.07.54.00.45"
    },
    {
      "Id": "GPU2",
      "State": "Enabled",
      "Manufacturer": "NVIDIA",
      "Model": "NVIDIA A16",
      "Serial Number": "1321023001692",
      "Firmware Version": "94.07.54.00.45"
    }
  ],
  "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_3BFC	Enabled
AES_NI_3C31	Enable
ARISupport_00E3	Enabled
AVXP1_3CBA	Nominal
Above4GDecoding_00DD	Enabled
AdjacentCachePrefetch_3C14	Enable
AdministratorPassword_0159	null
BitsPerSecond_0081	115200
BitsPerSecond_008B	115200
BootURI_292F	null
BootupNumLockState_0029	On
BusMasterEnable_00E4	Enabled
CPUC1AutoDemotion_3CDC	Enable
CPUC6Report_3CDE	Auto
CSMSupport_0163	Enabled
ChangeSettings_006E	Auto
ChangeSettings_0070	Auto
ClientPassword_2942	null
ClientUserName_014E	null
ConsistentDeviceNameSupport_00E5	Disabled
ConsoleRedirectionEMS_0077	0

4.3.2 Exporting BIOS Configurations

Use the command to export a BIOS configuration file in JSON format. You 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 file format, 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
-----	-----	-----
ADDDCSparing_3C69	Enabled	Disabled
OptionROMMessages_0028	Keep Current	Force BIOS

```
- update attributes
```

```
- BIOS Configuration Pending Settings
```

These settings will be applied on next system reboot

Item	Value
-----	-----
ADDDCSparing_3C69	Disabled
OptionROMMessages_0028	Force BIOS

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

To update the BIOS firmware, run the command `bios fw update -f <file>`. After the update is finished, the system must be rebooted for the update to take effect.



Notes:

- If an error occurs during the update process, the firmware update will be cancelled and exit the update mode.
- If the firmware is already in the update mode, it might be due to a previously unfinished update. The command will cause an error and exit the firmware 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.
- **--overwrite_smbios** option is used to overwrite SMBIOS region. Unless you are familiar with smbios data, do not use this option.
- **--backup** option is used to back up the active image during BIOS FW update. This image will be used by CPLD to recover BIOS from the backup image. Note that this option only works on ROT platform.
- **--overwrite_oa** option is used to overwrite the operating system information for Tatlow and later platforms.
- **--overwrite_setup_conf** option is used to overwrite the BIOS Setup configuratoin

-

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password>
    bios fw update -f <file>
    [--preserve_nv]
    [--preserve_mer]
    [--overwrite_smbios]
    [--backup]
    [--overwrite_oa]
    [--overwrite_setup_conf]
    [--overwrite_setup_pwd]
    [--overwrite_secure_bootkeys]
    [--overwrite boot conf]
```

```
optional arguments:
  -f FILE, --file FILE          file to read
  -s FILE, --source FILE        file to read source code from
  -t FILE, --target FILE        file to write target code to
  -o FILE, --output FILE        file to write output to
  -v, --verbose                 verbose output
  -h, --help                    show this help message
  --version                     show version
```

```

-t <file>, --file <file>      firmware image
--preserve_nv                  reserve NVRAM data
--preserve_mer                  reserve ME firmware region
--overwrite_smbios              overwrite SMBIOS region
--backup                        back up the active image (ROT only)
--overwrite_oa                  overwrite OA
--overwrite_setup_conf          overwrite the BIOS Setup Configuration
--overwrite_setup_pwd           overwrite the BIOS Setup password
--overwrite_secure_bootkeys     overwrite the BIOS Secure Boot Keys
--overwrite_boot_conf           overwrite the BIOS Boot Options Configuration

```

Example:

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

Console Output:

Here is an example of updating BIOS Firmware on X12.

- ```
1. Check if an image exists.
2. Uploading image and update firmware... Please wait, this may take a while!
[>>] 100.0%
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

To update BMC firmware, run the command `bmc fw update -f <file>`. After the update is finished, it will take a few minutes for the BMC to reset.



**Notes:**

- If an error occurs during the update process, the firmware update will be cancelled and exit the update mode.
- If the firmware is already in the update mode, it might be due to a previously unfinished update. The command will cause an error and exit the firmware update mode. You need to run the command again to update the firmware.
- You have to reboot or power up the managed system for getting BIOS information correctly after finishing BMC firmware update.
- **--overwrite\_cfg** option is used to restore the factory default BMC settings.
- **--overwrite\_sdr** option is used to restore the default SDR settings.
- **--overwrite\_sslcert** option is used to restore the default SSL certificate.
- **--backup** option is used to back up the active image during BMC FW update. This image will be used by CPLD to recover BMC from the backup image. Note that this option only works on ROT platform.

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password>
 bmc fw update -f <file>
 [--overwrite_cfg]
 [--overwrite_sdr]
 [--overwrite_sslcert]
 [-backup]
```

optional arguments:

[illegible]

Example:

```
./nagios_supermicro_redfish -i10.136.33.92 -uADMIN -pADMIN bmc fw update -f
~/firmware/REDFISH X10 369.bin
```

### Console Output:

Here is an example of updating BMC Firmware on X12.

- ```

1. Check if an image exists.
2. Uploading image and update firmware... Please wait, this will take a while!
[>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>] 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 users to create subscriptions to receive events. The user 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 sent 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	Alert, StatusChange
Context	public
Status	Enabled
----	----
Id	2
Destination	http://10.147.34.93:8080
Event Types	Alert, ResourceAdded, ResourceRemoved, ResourceUpdated, StatusChange
Context	public
Status	Enabled
----	----

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]]
[--id <id>]
[--status [STATUS_TYPE]]
```

positional arguments:

`<dest>` URL of destination of destination(`http{s}://<ip>:<port>`)

optional arguments:

- `--type [EVENT_TYPE]` event type: `StatusChange`, `ResourceUpdated`, `ResourceAdded`, `ResourceRemoved` and `Alert` (default: `StatusChange`, `Alert`)
- `--context [CONTEXT]` context (default: `public`)
- `--id <id>` Subscription id (The `--id` value ranges from 1 to 16. Note the option is only available on X12 and later platforms.)
- `--status [STATUS_TYPE]` status type: `enabled`, `disabled` (Note the option is only available on X12 and later platforms.)

Example:

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

Adding a subscription on X12 and later platforms.

```
./nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN event-service add https://10.136.161.69:8443 --type StatusChange ResourceUpdated ResourceAdded ResourceRemoved Alert --context private --id 5
```

Adding a disabled subscription on X12 and later platforms.

```
./nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN event-service add http://10.136.161.69:8080 --type StatusChange ResourceUpdated ResourceAdded ResourceRemoved Alert --context private --id 5 --status disabled
```

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
```

4.6 Log Service

The log service is used to get the information of health event log or maintenance event log.

4.6.1 Getting Health Event Log Information

Usage:

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

Example:

```
$/nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN log info
```

Console Output:

Name	Health Event Log Service
DateTime	2023-11-23 02:44:50
Enabled	True
Log count	65/4096

4.6.2 Getting Information of Maintenance Event Log

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> log info -t syslog
```

Example:

```
$/nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN log info -t syslog
```

Console Output:

Name	Maintenance Event Log Service
DateTime	2023-11-23 02:43:50
Enabled	True
Log count	187/512

4.6.3 Getting Health Event Logs

Usage:

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

Example:

```
$/nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN log list
```

Console Output:

```
#1 Critical, 2020-09-02 02:14:57, Physical security, [ Chassis Intru ] General
Chassis Intrusion
#2 Critical, 2020-09-02 02:15:00, Power supply, [ PS2 Status ] Power Supply
Failure detected
#3 Warning, 2020-09-02 02:31:13, None, [ OEM ] First AC Power on
#4 Critical, 2020-09-02 02:32:40, Physical security, [ Chassis Intru ] General
Chassis Intrusion
#5 Critical, 2020-09-02 02:32:44, Power supply, [ PS2 Status ] Power Supply
Failure detected
#6 Critical, 2020-09-02 06:29:22, Physical security, [ Chassis Intru ] General
Chassis Intrusion
#7 Critical, 2020-09-02 06:29:25, Power supply, [ PS2 Status ] Power Supply
Failure detected
```

4.6.4 Getting Maintenance Event Logs

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> log list -t syslog
```

Example:

```
$/nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN log list -t syslog
```

Console Output:

```
#1 2020-09-02 02:05:35, Redfish, ADMIN(ADMIN) [None] - IPMI configuration was
restored to default successfully.
#2 2020-09-02 02:05:35, Redfish, ADMIN(ADMIN) [None] - SSL certificate was
cleared successfully.
#3 2020-09-02 02:05:35, Redfish, ADMIN(ADMIN) [None] - BMC was reset
successfully.
#4 2020-09-02 02:13:19, DRTM, ADMIN(ADMIN) [None] - ID 0x00 - TEE FW Start
(0000.00.14)
#5 2020-09-02 02:13:19, DRTM, ADMIN(ADMIN) [None] - ID 0x01 - SMCI_TEE_SERVICE
(STS) Start
#6 2020-09-02 02:14:56, DRTM, ADMIN(ADMIN) [None] - ID 0x00 - TEE FW Start
(0000.00.14)
#7 2020-09-02 02:14:57, DRTM, ADMIN(ADMIN) [None] - ID 0x01 - SMCI_TEE_SERVICE
(STS) Start
#8 2020-09-02 02:14:58, DRTM, ADMIN(ADMIN) [None] - ID 0x02 - Security Functions
Start (TAS)
#9 2020-09-02 02:14:59, DRTM, ADMIN(ADMIN) [None] - ID 0x02 - Security Functions
Start (TA0)
```

```
#10 2020-09-02 02:15:00, DRTM, ADMIN(ADMIN) [None] - ID 0x02 - Security Functions
Start (TA3)
#11 2020-09-02 02:15:28, RMCP, ADMIN(ADMIN) [None] - Event subscription #2's
destintation was configured to 10.147.33.41 successfully.
#12 2020-09-02 02:15:41, DRTM, ADMIN(ADMIN) [None] - ID 0x02 - Security Functions
Start (TA1)
#13 2020-09-02 02:15:42, DRTM, ADMIN(ADMIN) [None] - ID 0x02 - Security Functions
Start (TA2)
```

4.6.5 Limiting the Number of Output Event Logs

To set the maximum number of latest event log entries to display, you can use the log list command with the “-l” parameter.

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> log list -l <count>
```

Example:

```
$/nagios_supermicro_redfish -i10.132.160.141 -uADMIN -pADMIN log list -l 5
```

Console Output:

```
#9 Critical, 2020-09-02 08:43:55, Power supply, [ PS2 Status ] Power Supply
Failure detected
#10 Critical, 2020-09-03 03:36:53, Physical security, [ Chassis Intru ] General
Chassis Intrusion
#11 Critical, 2020-09-03 03:36:56, Power supply, [ PS2 Status ] Power Supply
Failure detected
#12 OK , 2020-09-04 02:49:24, Physical security, [ Chassis Intru ] General
Chassis Intrusion
#13 Critical, 2020-09-04 03:20:05, Power supply, [ PS2 Status ] Power Supply
Failure detected
```

4.7 Firmware Inventory Management

Use this command to perform firmware inventory actions. The administrator can manually recover BMC or BIOS from the backup image. User can also update the current active image as golden template. This golden image will be utilized as a second option for recovery by CPLD. Note: Firmware inventory update commands are applicable only on the Root of Trust (ROT) platform.

4.7.1 Listing Firmware Inventory Information

Usage:

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

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN fw_inventory info
```

Console Output:

Name	Version
----	----
BMC	01.03.04
BMC Backup	01.03.04
BMC Golden	01.03.09
BMC Staging	01.03.04
BIOS	BIOS Date: 06/29/2022 Ver 1.3
BIOS Backup	BIOS Date: 06/29/2022 Ver 1.3
BIOS Golden	BIOS Date: 06/29/2022 Ver 1.3
BIOS Staging	BIOS Date: 06/29/2022 Ver 1.3
CPLD Motherboard	F1.00.01
BIOS ME	4.4.4.62
NIC1 System Slot0	None
Power supply 1	1.3
Power supply 2	1.0

4.7.2 Updating BMC Firmware Golden Image

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> fw_inventory update bmc --update_golden
```

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN fw_inventory update bmc --update_golden
```

Console Output:

```
Command successful. BMC may be disconnected during the update.  
Please wait for BMC to complete its recovery.
```

```
Update is done.
```

4.7.3 Recovering BMC Firmware

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> fw_inventory update bmc --recover
```

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN fw_inventory update bmc --recover
```

Console Output:

```
Command successful. BMC may be disconnected during the update.  
Please wait for BMC to complete its recovery.
```

```
Update is done.
```

4.7.4 Updating BIOS Firmware Golden Image

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> fw_inventory update bios --update_golden
```

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN fw_inventory update bios --update_golden
```

Console Output:

```
Command successful. BIOS update is in progress.  
Please wait, as BMC functionality may be limited during the update.
```

```
Update is done.
```

4.7.5 Recovering BIOS Firmware

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> fw_inventory update bios --recover
```

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN fw_inventory update bios --recover
```

Console Output:

```
Command successful. BIOS update is in progress.  
Please wait, as BMC functionality may be limited during the update.
```

```
Update is done.
```

4.9 Redfish Version

Display the current redfish version of the BMC.

4.9.1 Getting the Redfish Version

Usage:

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

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN redfish_ver
```

Console Output:

Name	Redfish
Version	1.9.0

4.10 Power control

Use this command to do power control.

4.10.1 Power status

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> power -t status
```

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN power -t status
```

Console Output:

Name	PowerState
Status	On

4.10.2 Power on

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> power -t on
```

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN power -t on
```

Console Output:

Done.

4.10.3 Power down immediately

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> power -t force_off
```

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN power -t force_off
```

Console Output:

Done.

4.10.4 Graceful shutdown

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> power -t graceful_shutdown
```

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN power -t graceful_shutdown
```

Console Output:

Done.

4.10.5 Power cycle

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> power -t cycle
```

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN power -t cycle
```

Console Output:

Done.

4.10.6 Power reset

Usage:

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

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN power -t reset
```

Console Output:

Done.

4.10.7 AC Power cycle

This option only works for X14/H14 and later platforms.

Usage:

```
nagios_supermicro_redfish -i <ip> -u <username> -p <password> power -t ac_cycle
```

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN power -t ac_cycle
```

Console Output:

Done.

4.11 BMC Redfish Raw Data

Use this command to retrieve the complete response from the specified BMC Redfish API path.

4.11.1 Get BMC redfish API Root

Usage:

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

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN raw
```

Console Output:

```
{
  "@odata.type": "#ServiceRoot.v1_9_0.ServiceRoot",
  "@odata.id": "/redfish/v1",
  "Id": "ServiceRoot",
  "Name": "Root Service",
  "RedfishVersion": "1.11.0",
  "UUID": "00000000-0000-0000-0000-7CC255181A1D",
  "Systems": {
    "@odata.id": "/redfish/v1/Systems"
  },
  "Chassis": {
    "@odata.id": "/redfish/v1/Chassis"
  },
  "Managers": {
    "@odata.id": "/redfish/v1/Managers"
  },
  "Tasks": {
    "@odata.id": "/redfish/v1/TaskService"
  },
  "SessionService": {
    "@odata.id": "/redfish/v1/SessionService"
  },
  "AccountService": {
    "@odata.id": "/redfish/v1/AccountService"
  },
  "EventService": {
    "@odata.id": "/redfish/v1/EventService"
  },
  ...
}
```

4.11.2 Get BMC redfish API

Usage:

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

Example:

```
$/nagios_supermicro_redfish -i 10.147.165.215 -u ADMIN -p ADMIN -api
/redfish/v1/Systems/1
```

Console Output:

```
{
  "@odata.type": "#ComputerSystem.v1_14_0.ComputerSystem",
  "@odata.id": "/redfish/v1/Systems/1",
  "Id": "1",
  "Name": "System",
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "Manufacturer": "Supermicro",
  "Model": "Super Server",
  "UUID": "00000000-0000-0000-0000-7CC255181A1D",
  "ProcessorSummary": {
    "Count": 1,
    "Model": "Intel(R) Xeon(R) processor"
  },
  "MemorySummary": {
    "TotalSystemMemoryGiB": 16
  },
  "PowerState": "On",
  "Boot": {
    "BootSourceOverrideEnabled": "Disabled",
    "BootOrder": [
      "Boot0004",
      "Boot0005"
    ]
  },
  ...
  "Actions": {
    "#ComputerSystem.Reset": {
      "target": "/redfish/v1/Systems/1/Actions/ComputerSystem.Reset"
    }
  }
}
```

5. Event Receiver

The Redfish server will send the event to the event receiver that has subscribed to event service. The Supermicro event receiver is a simple web server, and it receives and writes events to a log file by a remote IP. Please note that this simple web server will accept data from any source. If there are security concerns, it is recommended to use iptables or other tools that can restrict the sources.

5.1 Supermicro Event Receiver

Usage:

```
supermicro_event_receiver [-h] -i <ip> [-p <port>] [--log_dir <log directory>]
                           [--protocol <http or https>]
                           [--cert <certificate.pem>]
                           [--key <privatekey.pem>]
```

optional arguments:

-h, --help	show this help message and exit
-i <ip>	event receiver IP
-p <port>	event receiver port, port range 1024-65535
--log_dir <log directory>	event log directory
--protocol <http{s}>	http or https receiver (default: http)
--cert <certificate.pem>	PEM format SSL certificate for https protocol
--key <privatekey.pem>	PEM format private key for https protocol

Example:

Here is an example of launching an HTTP receiver.

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

Console Output:

```
supermicro_event_receiver 1.1.0 build 201230
```

```
Log directory: /home/supermicro/nagios_supermicro_redfish/bin/log
```

```
* Serving Flask app 'event_receiver'
```

```
* Debug mode: off
```

```
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
```

```
* Running on http://10.136.160.119:8080
```

```
Press CTRL+C to quit.
```

Here is an example of launching an HTTPS receiver.

```
$ ./supermicro_event_receiver -i 10.147.1.2 --protocol https
```

Console Output:

```
supermicro_event_receiver 1.3.0 build 240507
```

```
Log directory: /home/supermicro/nagios_supermicro_redfish/bin/log
```

```
* Serving Flask app 'event_receiver'
```

```
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on https://10.147.1.2:8443
Press CTRL+C to quit.
```

5.2 Event Log

An event is written to a log file. The filename format is `event_<remote ip>.log`

The format of an event log:

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

Appendix A Third-Party Software

The following open source libraries are used in the Nagios plugin package.

Library	License
requests	Apache 2.0
Requests-toolbelt	Apache 2.0
Flask	BSD

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)
Sales-USA@supermicro.com (Sales Inquiries)
Government_Sales-USA@supermicro.com (Gov. Sales Inquiries)
support@supermicro.com (Technical Support)
RMA@supermicro.com (RMA Support)

Website: 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_Europe@supermicro.com (Sales Inquiries)
Support_Europe@supermicro.com (Technical Support)
RMA_Europe@supermicro.com (RMA Support)

Website: www.supermicro.nl

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

Email: Sales-Asia@supermicro.com.tw (Sales Inquiries)
Support@supermicro.com.tw (Technical Support)
RMA@supermicro.com.tw (RMA Support)

Website: www.supermicro.com.tw