



NVIDIA Switch BMC User Manual

v88.0002.1301

Table of Contents

1	Overview	7
1.1	Intended Audience	7
1.2	Baseboard Management Controller (BMC).....	7
1.2.1	High-Level Feature List	8
1.2.2	BMC-to-CPU Communication	8
1.2.3	Firmware Upgrade	8
1.3	Terminology.....	9
1.4	System Features.....	9
2	Getting Started.....	11
2.1	Prerequisites.....	11
2.2	Getting Started.....	11
2.2.1	Login Credentials	11
2.2.2	Serial Console Management	12
2.2.3	Wired Ethernet Management.....	12
3	User Interface Redfish Commands	13
3.1	Firmware Management.....	15
3.1.1	Show Firmware Inventory.....	15
3.1.2	Show Firmware Version & Health.....	16
3.1.3	Update Firmware	17
3.1.4	Show Update Firmware Status	18
3.1.5	Filter Next Update Firmware End Components	20
3.1.6	Updating Firmware with Multipart API	21
3.1.7	Show Update Firmware Multipart Status	22
3.1.8	Firmware Update Expected Duration	24
3.2	Chassis.....	24
3.2.1	Show Chassis Information.....	24
3.2.2	Show Chassis Component Information	25
3.3	Certificate Service.....	26
3.3.1	Show Certificate Service.....	27
3.3.2	Show Certificate Locations.....	27
3.3.3	Show Certificate	28
3.4	Session Service	29

3.4.1	Show Session Service	29
3.4.2	Show Sessions.....	29
3.4.3	Show Session Details.....	30
3.5	User Management	30
3.5.1	Show BMC User Account Configuration	31
3.5.2	Show BMC User Accounts	33
3.5.3	Create New BMC User	33
3.5.4	Change BMC User Password.....	34
3.5.5	Change BMC "root" User Account Password from BMC "admin" User Account ..	35
3.5.6	Change BMC User Account Permissions	35
3.5.7	Change BMC Account Lockout Duration.....	36
3.5.8	Change BMC Minimum Password Length.....	37
3.5.9	Change BMC Account Lockout Threshold	38
3.5.10	Show BMC Account Service Roles	38
3.5.11	Show BMC User Account Service Role Options.....	39
3.6	Attestation	40
3.6.1	Show System ERoT List	40
3.6.2	Show ERoT Security Information	41
3.6.3	Generate ERoT SPDM Information.....	42
3.6.4	Show ERoT SPDM Generation Status	43
3.6.5	Show ERoT SPDM Information.....	44
3.6.6	Set ERoT Automatic Background Copy State	45
3.6.7	Show ERoT Automatic Background Copy State	46
3.6.8	Show Minimum Security Version Information for Application Firmware	47
3.7	Power Management	49
3.7.1	Apply CPU Host Reset	49
3.7.2	CPU Host Reset Action Information	50
3.7.3	Apply BMC Manager Reset	50
3.7.4	BMC Host Reset Action Information	51
3.8	Time Management	52
3.8.1	Show NTP Servers Status.....	52
3.8.2	Enable/Disable NTP Servers	53
3.8.3	Show Time and Date.....	53
3.8.4	Set Time and Date Manually.....	55

3.9	Network Interfaces	55
3.9.1	Show Network Interface List	56
3.9.2	Show Network Interface Details	56
3.9.3	Set BMC Hostname	57
3.10	EEPROM	59
3.10.1	Show EEPROM Information	59
3.11	Temperature Sensor	60
3.11.1	Show Temperature Sensor Information	60
3.12	Service Identification	61
3.12.1	Set Service Identification	61
3.12.2	Show Service Identification	62
3.13	Debug Information	64
3.13.1	Generate Debug Information	64
3.13.2	Show Debug Information Generation Status	65
3.13.3	Output Debug Information To File	66
3.14	Debug Token (CRDT)	67
3.14.1	Generate Debug Token	67
3.14.2	Show Debug Token Generation Status	68
3.14.3	Download Debug Token	70
3.14.4	Install Debug Token Signed Firmware	70
3.14.5	Show Debug Token Installation Status	71
3.14.6	Generate Installed Debug Token Attachments	72
3.14.7	Show Installed Debug Token Attachments Generation Status	72
3.14.8	Show Installed Debug Token Attachments	75
3.15	Leakage Sensor	76
3.15.1	Register to Leakage Events	76
3.15.2	Leakage Sensor Status	77
3.16	Factory Reset	77
3.16.1	Factory Reset (Configuration Only)	77
3.16.2	Factory Reset (Configuration & Logs)	78
3.17	eMMC Secure Erase	79
3.17.1	Securely Erase eMMC Card	79
3.18	Rsyslog	79
3.18.1	Configure Rsyslog Client	80

3.18.2	Query Rsyslog Client Configuration	80
3.18.3	Enable Encrypted Streaming With TLS.....	81
3.18.4	Change Rsyslog Transport Protocol.....	82
3.18.5	Configure Facility Filters	82
3.18.6	Configure Priority Filters	83
3.19	Supported Redfish Scheme URIs	84
4	Document Revision History	87

NVIDIA Yocto Baseboard Management Controller (BMC) operating system enables the chassis management and configuration of NVIDIA's GB200 NVLink switch tray (N5xxx_LD) platform. This operating system provides a suite of management options, incorporates API, which enables administrators to easily configure and manage the system chassis management.

These pages provide information about the scope, organization, and command-line interface of NVIDIA Yocto BMC operating system as well as configuration examples.

Document Revision History

A list of the changes made to the User Manual are provided in [Document Revision History](#).

1 Overview

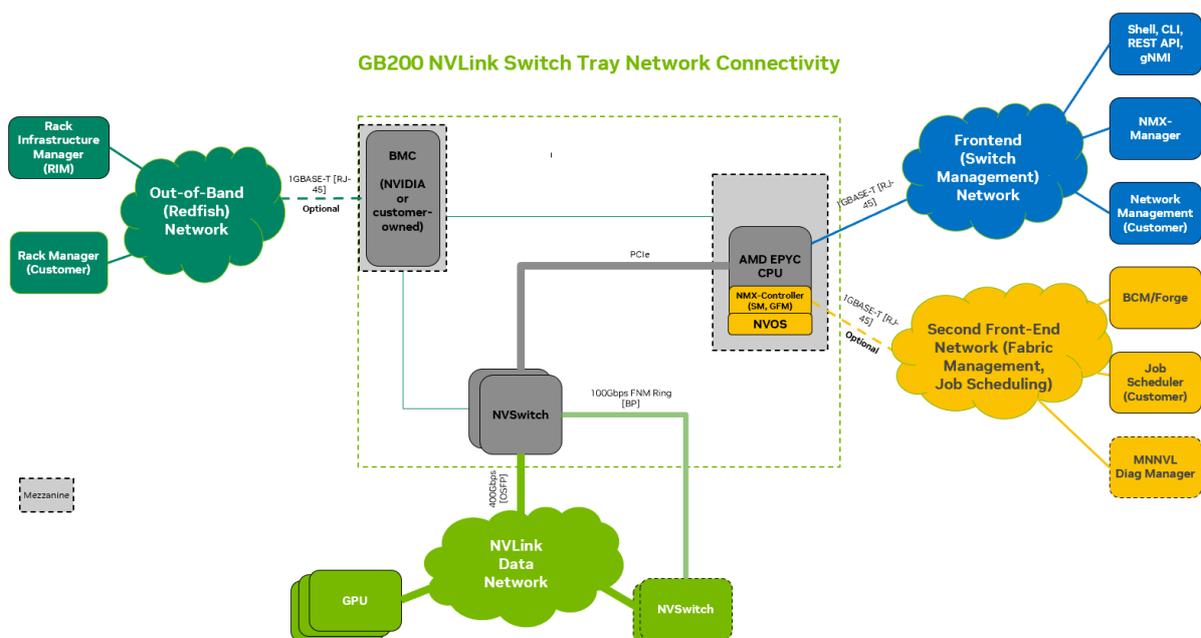
- [1.1 Intended Audience](#)
- [1.2 Baseboard Management Controller \(BMC\)](#)
 - [1.2.1 High-Level Feature List](#)
 - [1.2.2 BMC-to-CPU Communication](#)
 - [1.2.3 Firmware Upgrade](#)
- [1.3 Terminology](#)
- [1.4 System Features](#)

1.1 Intended Audience

These pages are intended for network administrators who are responsible for configuring and managing BMC-based platforms. The instructions in this guide presuppose a moderate understanding of Linux, encompassing skills such as text editing, comprehending Unix file permissions, and process monitoring.

1.2 Baseboard Management Controller (BMC)

In the GB200 NVLink switch tray systems, the BMC (Baseboard Management Controller) ensures that the management plane is distinct from the control plane, a requirement of some Cloud Service Providers (CSPs). It's an essential server element offering hardware monitoring and security capabilities. Within the GB200 switch tray, the BMC operates alongside NVOS, which is the operating system on the host CPU managing the system, connecting via Ethernet over USB. The BMC manages most system peripherals, and the customer engages with it through a dedicated 1GbE RJ-45 interface utilizing the Redfish protocol. For those foregoing BMC network connections, its functionalities are also accessible via NVOS UI.



1.2.1 High-Level Feature List

- Platform attestation (for all ERoTs)
- Update firmware using PLDM T5 via ERoT, including the following firmware components:
 - BMC software/firmware
 - FPGA
 - CPU BIOS (UEFI and NVOS is updated from CPU SSD)
 - NVSwitch firmware (launched ONLY from NVOS as part as NVOS upgrade)

For full list of supported commands and events, please see the [System Features](#) section, below.

1.2.2 BMC-to-CPU Communication

The CPU communicates with the BMC using Redfish over Ethernet over USB. BMC reads information from the CPU using MCTP over I²C.

In every NVIDIA platform that facilitates local communication between the BMC and CPU, static private IPv4 addresses are configured.

- BMC side 10.0.1.1 (netmask 255.255.252.0)
- CPU side 10.0.1.2 (netmask 255.255.252.0)

The Redfish client on the CPU will be bound to the USB I/F, thus ensuring that internal traffic cannot "leak" to the management network. A dedicated `nvos_user` SHALL be added to the BMC image. When NVOS connects to BMC it SHALL change the password to a unique password based on SEED from the CPU TPM. If CPU fails to connect, it reverts to the default password. Each entity MUST be capable of operating without the other one, meaning the following:

- BMC SHALL operate normally while CPU is undergoing reboot; BMC must detect the connectivity state towards the CPU.
- CPU SHALL operate normally while BMC is undergoing reboot; CPU must detect the connectivity state towards the BMC.

1.2.3 Firmware Upgrade

- BMC upgrades via the relevant ERoT: BMC firmware, FPGA, CPU BIOS, ERoTs firmware and NVSwitch firmware, and upgrades CPLD directly (From CPU not from BMC).
- The flow is PLDM Type 5, writing to staging FLASH. Then ERoT validates the integrity and continues the flow.
- The trigger for firmware update can be done either with external Redfish or NVOS.
- NVOS SHALL utilize MFT for the actual process through BMC.
- NVSwitch firmware upgrade MUST be done via NVOS to ensure compatibility with the driver and SDK version.

1.3 Terminology

Term	Description
ASIC	An application-specific integrated circuit (ASIC) is an integrated circuit (IC) chip customized for a particular use, rather than intended for general-purpose use.
Attestation	Authenticated, machine-readable metadata about one or more software artifacts. An attestation MUST contain at least: Envelope.
BMC	Baseboard Management Controller
DMTF	Founded in 1992, DMTF (formerly known as the Distributed Management Task Force) is an industry standards organization working to simplify the manageability of network-accessible technologies through open and collaborative efforts by leading technology companies.
EEPROM	A type of non-volatile ROM that enables individual bytes of data to be erased and reprogrammed.
EROT	External root of trust
FPGA	Field Programmable Gate Arrays (FPGAs) are integrated circuits often sold off-the-shelf. They're referred to as 'field programmable' because they provide customers the ability to reconfigure the hardware to meet specific use case requirements after the manufacturing process.
Host	A computer platform executing an Operating System which may control one or more network adapters. A host in networking is a device that is connected to a network and is able to communicate with other hosts on the network.
Network Interfaces	A <i>network interface</i> is the point of interconnection between a computer and a private or public network. A network interface is generally a network interface card (NIC), but does not have to have a physical form. Instead, the network interface can be implemented in software.
RF	DMTF's <i>Redfish</i> ® is a standard designed to deliver simple and secure management for converged, hybrid IT and the Software Defined Data Center (SDDC).
SPDM	DMTF's Security Protocol and Data Model (SPDM) Specification defines messages, data objects, and sequences for performing message exchanges between devices over a variety of transport and physical media to authentication of components, firmware measurement and protection of data in flight.
VPD	Vital product data

1.4 System Features

Feature	Detail
Firmware Management	<ul style="list-style-type: none"> • Show Firmware Inventory • Show Firmware Version & Health • Update Firmware • Show Firmware Update Status • Filter Next Update Firmware End Components
User Management	<ul style="list-style-type: none"> • Show BMC Users • Change BMC User Password

Feature	Detail
Attestation	<ul style="list-style-type: none"> • Show System EROT List • Show EROT Security Information • Generate EROT SPDM Information • Show EROT SPDM Generation Status • Show EROT SPDM Information • Set EROT Automatic Background Copy State • Show EROT Automatic Background Copy Status
Power Management	<ul style="list-style-type: none"> • Apply Reset
Network Interfaces	<ul style="list-style-type: none"> • Show Network Interface List • Show Network Interface Details
EEPROM	<ul style="list-style-type: none"> • Show EEPROM Information
Temperature Sensor	<ul style="list-style-type: none"> • Show Temperature Sensor Information
Debug Information	<ul style="list-style-type: none"> • Generate Debug Information • Show Debug Information Generation Status • Output Debug Information To File
Debug Token	<ul style="list-style-type: none"> • Generate Debug Tokens • Show Debug Token Generation Status • Download Debug Token • Install Debug Token Signed Firmware • Show Debug Token Installation Status • Generate Installed Debug Token Attachments • Show Installed Debug Token Attachments Generation Status • Show Installed Debug Token Attachments
Leakage Sensor	<ul style="list-style-type: none"> • Show the status of the Leak Detect sensors • Leakage notification VIA RF events
Rsyslog	<ul style="list-style-type: none"> • The BMC can stream out local logs (that go to the systemd journal) by using rsyslog.

2 Getting Started

- [2.1 Prerequisites](#)
- [2.2 Getting Started](#)
 - [2.2.1 Login Credentials](#)
 - [2.2.2 Serial Console Management](#)
 - [2.2.3 Wired Ethernet Management](#)

This section provides an end-to-end setup process for installing and running BMC.

2.1 Prerequisites

- A variety of text editors are pre-installed, including `vi` and `nano`.
- Access to a Linux or UNIX shell is needed. For Windows users, employing a Linux environment like `Cygwin` as the command line tool is recommended for interacting with BMC.
- There must be some kind of management port allowing access to the BMC controller (either via host, USB, or direct BMC management port).

2.2 Getting Started

When starting BMC for the first time, the management port sends a DHCP request. To determine the IP address of the switch's BMC management port, the MAC address of the switch's BMC can be cross referenced with the DHCP server.

The MAC address is typically located on the side of the switch tray or on the box in which the unit ships.

2.2.1 Login Credentials

The default installation includes two accounts which have full system privileges:

1. The system account (`root`). The root account uses the default password `openBmc`.

 It is mandatory to change the default root password when logging in for the first time. Refer to the "Change BMC User Password" RF command.

2. The user account (`admin`).

 While the admin user exists, it should not be deleted nor renamed and its password should not be changed.

In this quick start guide, the `root` account is used to configure the BMC.

All accounts can use remote SSH login. Note that the SSH auto-logout is 15 minutes.

2.2.2 Serial Console Management

The serial console is directed by default to the host and can be modified to show the BMC output by issuing a command.

```
# from within BMC's shell
BMC_TO_CPU_UART=/sys/devices/platform/ahb/ahb:apb/ahb:apb:bus@1e78a000/1e78a300.i2c-bus/i2c-5/5-0031/mlxreg-io/
hwmon/hwmon*/uart_sel
# Switch UART to CPU.
echo 1 > $BMC_TO_CPU_UART
# Switch UART to BMC.
echo 0 > $BMC_TO_CPU_UART
```

It is recommended to perform management and configuration over the network, either in-band or out-of-band. A serial console is fully supported.

The standard baud rate for the serial console is 115200, with an RJ45 as the physical connection interface.

2.2.3 Wired Ethernet Management

A BMC always provides two dedicated Ethernet management ports called "eth0" and "usb0". The interface eth0 is specifically for out-of-band management use and the usb0 is the in-band interface through which the CPU may interact with the BMC. By default, the management interface uses DHCP for addressing.

3 User Interface Redfish Commands

- [3.1 Firmware Management](#)
 - [3.1.1 Show Firmware Inventory](#)
 - [3.1.2 Show Firmware Version & Health](#)
 - [3.1.3 Update Firmware](#)
 - [3.1.4 Show Update Firmware Status](#)
 - [3.1.5 Filter Next Update Firmware End Components](#)
 - [3.1.6 Updating Firmware with Multipart API](#)
 - [3.1.7 Show Update Firmware Multipart Status](#)
 - [3.1.8 Firmware Update Expected Duration](#)
- [3.2 Chassis](#)
 - [3.2.1 Show Chassis Information](#)
 - [3.2.2 Show Chassis Component Information](#)
- [3.3 Certificate Service](#)
 - [3.3.1 Show Certificate Service](#)
 - [3.3.2 Show Certificate Locations](#)
 - [3.3.3 Show Certificate](#)
- [3.4 Session Service](#)
 - [3.4.1 Show Session Service](#)
 - [3.4.2 Show Sessions](#)
 - [3.4.3 Show Session Details](#)
- [3.5 User Management](#)
 - [3.5.1 Show BMC User Account Configuration](#)
 - [3.5.2 Show BMC User Accounts](#)
 - [3.5.3 Create New BMC User](#)
 - [3.5.4 Change BMC User Password](#)
 - [3.5.5 Change BMC "root" User Account Password from BMC "admin" User Account](#)
 - [3.5.6 Change BMC User Account Permissions](#)
 - [3.5.7 Change BMC Account Lockout Duration](#)
 - [3.5.8 Change BMC Minimum Password Length](#)
 - [3.5.9 Change BMC Account Lockout Threshold](#)
 - [3.5.10 Show BMC Account Service Roles](#)
 - [3.5.11 Show BMC User Account Service Role Options](#)
- [3.6 Attestation](#)
 - [3.6.1 Show System ERoT List](#)
 - [3.6.2 Show ERoT Security Information](#)
 - [3.6.3 Generate ERoT SPDM Information](#)
 - [3.6.4 Show ERoT SPDM Generation Status](#)
 - [3.6.5 Show ERoT SPDM Information](#)
 - [3.6.6 Set ERoT Automatic Background Copy State](#)
 - [3.6.7 Show ERoT Automatic Background Copy State](#)
 - [3.6.8 Show Minimum Security Version Information for Application Firmware](#)
- [3.7 Power Management](#)
 - [3.7.1 Apply CPU Host Reset](#)
 - [3.7.2 CPU Host Reset Action Information](#)
 - [3.7.3 Apply BMC Manager Reset](#)

- [3.7.4 BMC Host Reset Action Information](#)
- [3.8 Time Management](#)
 - [3.8.1 Show NTP Servers Status](#)
 - [3.8.2 Enable/Disable NTP Servers](#)
 - [3.8.3 Show Time and Date](#)
 - [3.8.4 Set Time and Date Manually](#)
- [3.9 Network Interfaces](#)
 - [3.9.1 Show Network Interface List](#)
 - [3.9.2 Show Network Interface Details](#)
 - [3.9.3 Set BMC Hostname](#)
- [3.10 EEPROM](#)
 - [3.10.1 Show EEPROM Information](#)
- [3.11 Temperature Sensor](#)
 - [3.11.1 Show Temperature Sensor Information](#)
- [3.12 Service Identification](#)
 - [3.12.1 Set Service Identification](#)
 - [3.12.2 Show Service Identification](#)
- [3.13 Debug Information](#)
 - [3.13.1 Generate Debug Information](#)
 - [3.13.2 Show Debug Information Generation Status](#)
 - [3.13.3 Output Debug Information To File](#)
- [3.14 Debug Token \(CRDT\)](#)
 - [3.14.1 Generate Debug Token](#)
 - [3.14.2 Show Debug Token Generation Status](#)
 - [3.14.3 Download Debug Token](#)
 - [3.14.4 Install Debug Token Signed Firmware](#)
 - [3.14.5 Show Debug Token Installation Status](#)
 - [3.14.6 Generate Installed Debug Token Attachments](#)
 - [3.14.7 Show Installed Debug Token Attachments Generation Status](#)
 - [3.14.8 Show Installed Debug Token Attachments](#)
- [3.15 Leakage Sensor](#)
 - [3.15.1 Register to Leakage Events](#)
 - [3.15.2 Leakage Sensor Status](#)
- [3.16 Factory Reset](#)
 - [3.16.1 Factory Reset \(Configuration Only\)](#)
 - [3.16.2 Factory Reset \(Configuration & Logs\)](#)
- [3.17 eMMC Secure Erase](#)
 - [3.17.1 Securely Erase eMMC Card](#)
- [3.18 Rsyslog](#)
 - [3.18.1 Configure Rsyslog Client](#)
 - [3.18.2 Query Rsyslog Client Configuration](#)
 - [3.18.3 Enable Encrypted Streaming With TLS](#)
 - [3.18.4 Change Rsyslog Transport Protocol](#)
 - [3.18.5 Configure Facility Filters](#)
 - [3.18.6 Configure Priority Filters](#)

3.1 Firmware Management

The firmware management section is composed of the following APIs, some of which work in conjunction:

- [Show Firmware Inventory](#)
- [Show Firmware Version & Health](#)
- [Update Firmware](#)
- [Show Update Firmware Status](#)
- [Filter Next Update Firmware End Components](#)
- [Updating Firmware with Multipart API](#)
- [Show Update Firmware Multipart Status](#)

The last three APIs are related to updating firmware components in the system.

The [Update Firmware](#) API initiates the task of updating the firmware of one or more firmware components in the system. Following that task initiation, one can call the [Show Update Firmware Status](#) API to check the state of that task (i.e., whether a task is still running or completed). In addition, before calling the [Update Firmware](#) API, one may choose to call the [Filter Next Update Firmware End Components](#) API and detail a component to be filtered and not updated, even if it is supplied to the [Update Firmware](#) API.

3.1.1 Show Firmware Inventory

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/UpdateService/FirmwareInventory	
Description	This Redfish API shall list all available firmware components: BMC, CPU, FPGA, and so forth.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	

Response Example	<pre> "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory", "@odata.type": "#SoftwareInventoryCollection.SoftwareInventoryCollection", "Members": [{ "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/CPLD_0" }, { "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/MGX_FW_BMC_0" }, { "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/MGX_FW_CPU_0" }, { "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/MGX_FW_ERoT_BMC_0" }, { "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/MGX_FW_ERoT_CPU_0" }, { "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/MGX_FW_ERoT_FPGA_0" }, { "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/MGX_FW_ERoT_NVSwitch_0" }, { "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/MGX_FW_ERoT_NVSwitch_1" }, { "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/MGX_FW_FPGA_0" },], Members@odata.count: 11, "Name": "Software Inventory Collection" </pre>
Related Commands	Show Firmware Version & Health Update Firmware Show ERoT Information
Notes	

3.1.2 Show Firmware Version & Health

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/UpdateService/FirmwareInventory/<comp_name>	
Description	This Redfish API shall list component and version information and health.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	comp_name	MGX_FW_BMC_0, MGX_FW_FPGA_0, and so forth (all components will have a meaningful version besides CPLD_0)
Default	N/A	
History	88.0002.0574	

Response Example	<pre> { "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/MGX_FW_BMC_0", "@odata.type": "#SoftwareInventory.v1_4_0.SoftwareInventory", "Description": "BMC image", "Id": "MGX_FW_BMC_0", "Manufacturer": "NVIDIA", "Name": "Software Inventory", "RelatedItem": [{ "@odata.id": "/redfish/v1/Chassis/MGX_BMC_0" }], "RelatedItem@odata.count": 1, "SoftwareId": "0x001B", "Status": { "Conditions": [], "Health": "OK", "HealthRollup": "OK", "State": "Enabled" }, "Updateable": true, "Version": "88.0002.0927", "WriteProtected": false } </pre>
Related Commands	<p>Show Firmware Inventory Update Firmware Show Update Firmware Status Filter Next Update Firmware End Components</p>
Notes	

3.1.3 Update Firmware



All firmware updates should be performed via NVOS. Only the BIOS firmware may be updated using this method, and only as a last resort if absolutely necessary.

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/octet-stream" -X POST https://<bmc_ip>/redfish/v1/UpdateService -T <image_name>.fwpkg	
Description	This Redfish API shall update the firmware image. Image format must be fwpkg format, suitable for PLDM. This Redfish API ignores any FW version checks, meaning one can upgrade or downgrade the FW version regardless of the current version.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	image_name	fwpkg image name
	id (return value)	task_id_number
Default	N/A	
History	88.0002.0574	

Response Example	<pre>{ "@odata.id": "/redfish/v1/TaskService/Tasks/0", "@odata.type": "#Task.v1_4_3.Task", "Id": "0", "TaskState": "Running", "TaskStatus": "OK" }</pre>
Related Commands	<p>Show Firmware Inventory Show Firmware Version & Health Show Update Firmware Status Filter Next Update Firmware End Components</p>
Notes	<ul style="list-style-type: none"> • The fwpkg file contains header for PLDM service • The header must include information about the desired component to be updated.

3.1.4 Show Update Firmware Status

Redfish API	curl -s -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/TaskService/Tasks/<task_id_number>	
Description	This Redfish API shall be used to get the Update Firmware task state. When task state changes from “Running” to “Completed”, the firmware is fully updated.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	task_id_number	Return value taken from Update Firmware command
Default	N/A	
History	88.0002.0574	

Response Example

```
"@odata.id": "/redfish/v1/TaskService/Tasks/0",
"@odata.type": "#Task.v1_4_3.Task",
"EndTime": "1970-01-01T00:30:39+00:00",
"Id": "0",
"Messages": [
  {
    "@odata.type": "#Message.v1_0_0.Message",
    "Message": "The task with id 0 has started.",
    "MessageArgs": [
      "0"
    ],
    "MessageId": "TaskEvent.1.0.1.TaskStarted",
    "Resolution": "None.",
    "Severity": "OK"
  },
  {
    "@odata.type": "#MessageRegistry.v1_4_1.MessageRegistry",
    "Message": "The resource property 'HGX_FW_Debug_Token_Erase' has detected errors of
type 'Device Discovery Failure'.",
    "MessageArgs": [
      "HGX_FW_Debug_Token_Erase",
      "Device Discovery Failure"
    ],
    "MessageId": "ResourceEvent.1.0.ResourceErrorsDetected",
    "Resolution": "Retry the firmware update operation and if issue still persists
reset the baseboard.",
    "Severity": "Critical"
  },
  {
    "@odata.type": "#MessageRegistry.v1_4_1.MessageRegistry",
    "Message": "Transfer of image '0.0' to 'HGX_FW_Debug_Token_Erase' failed.",
    "MessageArgs": [
      "0.0",
      "HGX_FW_Debug_Token_Erase"
    ],
    "MessageId": "Update.1.0.TransferFailed",
    "Resolution": "None.",
    "Severity": "Critical"
  },
  {
    "@odata.type": "#MessageRegistry.v1_4_1.MessageRegistry",
    "Message": "The target device '27' will be updated with image 'COMP_VERSION'.",
    "MessageArgs": [
      "27",
      "COMP_VERSION"
    ],
    "MessageId": "Update.1.0.TargetDetermined",
    "Resolution": "None.",
    "Severity": "OK"
  },
  {
    "@odata.type": "#MessageRegistry.v1_4_1.MessageRegistry",
    "Message": "Image 'COMP_VERSION' is being transferred to '27'.",
    "MessageArgs": [
      "COMP_VERSION",
      "27"
    ],
    "MessageId": "Update.1.0.TransferringToComponent",
    "Resolution": "None.",
    "Severity": "OK"
  },
  {
    "@odata.type": "#Message.v1_0_0.Message",
    "Message": "The task with id 0 has changed to progress 20 percent complete.",
    "MessageArgs": [
      "0",
      "20"
    ],
    "MessageId": "TaskEvent.1.0.1.TaskProgressChanged",
    "Resolution": "None.",
    "Severity": "OK"
  },
  {
    "@odata.type": "#Message.v1_0_0.Message",
    "Message": "The task with id 0 has changed to progress 40 percent complete.",
    "MessageArgs": [
      "0",
      "40"
    ],
    "MessageId": "TaskEvent.1.0.1.TaskProgressChanged",
    "Resolution": "None.",
    "Severity": "OK"
  },
  {
    "@odata.type": "#MessageRegistry.v1_4_1.MessageRegistry",
    "Message": "Device '27' successfully updated with image 'COMP_VERSION'.",
    "MessageArgs": [
      "27",
      "COMP_VERSION"
    ],
    "MessageId": "Update.1.0.UpdateSuccessful",
    "Resolution": "None.",
    "Severity": "OK"
  },
  {
    "@odata.type": "#MessageRegistry.v1_4_1.MessageRegistry",
    "Message": "Awaiting for an action to proceed with activating image 'COMP_VERSION'
on '27'.",
    "MessageArgs": [
      "COMP_VERSION",

```

	<pre> "27"], "MessageId": "Update.1.0.AwaitToActivate", "Resolution": "System reboot or AC power cycle", "Severity": "OK" }, { "@odata.type": "#Message.v1_0_0.Message", "Message": "The task with id 0 has changed to progress 100 percent complete.", "MessageArgs": ["0", "100"], "MessageId": "TaskEvent.1.0.1.TaskProgressChanged", "Resolution": "None.", "Severity": "OK" }, { "@odata.type": "#Message.v1_0_0.Message", "Message": "The task with id 0 has Completed.", "MessageArgs": ["0"], "MessageId": "TaskEvent.1.0.1.TaskCompletedOK", "Resolution": "None.", "Severity": "OK" } }], "Name": "Task 0", "Payload": { "HttpHeaders": ["Host: 10.0.1.1", "User-Agent: curl/7.74.0", "Accept: */*", "Content-Length: 67109132"], "HttpOperation": "POST", "JsonBody": "null", "TargetUri": "/redfish/v1/UpdateService" }, "PercentComplete": 100, "StartTime": "1970-01-01T00:19:46+00:00", "TaskMonitor": "/redfish/v1/TaskService/Tasks/0/Monitor", "TaskState": "Completed", "TaskStatus": "OK" </pre>
Related Commands	Show Firmware Inventory Show Firmware Version & Health Update Firmware Filter Next Update Firmware End Components
Notes	CPLD_0, NVSwitch_0, NVSwitch_1 are not included for N5110_LD system in version 88.0002.0472.

3.1.5 Filter Next Update Firmware End Components

Redfish API	curl -k -u <user>:<p/w> -X PATCH -d '{"HttpPushUriTargets":["/redfish/v1/UpdateService/FirmwareInventory/<comp_name>"]}' https://<bmc_ip>/redfish/v1/UpdateService	
Description	This Redfish API shall be used to apply a filter for a fwpkg file consisted of several firmware components.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	comp_name	MGX_FW_CPU_0 (only BIOS firmware upgrade is applicable from BMC).
Default	N/A	
History	88.0002.0574	

Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.16.0.Success", "MessageSeverity": "OK", "Resolution": "None" }] } </pre>
Related Commands	<p>Show Firmware Inventory Show Firmware Version & Health Update Firmware Show Update Firmware Status</p>
Notes	This command MUST be called before Update Firmware command in order to take effect.

3.1.6 Updating Firmware with Multipart API

Redfish API	<pre> curl -k -u <user>:<p/w> https://<bmc_ip>/redfish/v1/UpdateService/update-multipart --form 'UpdateParameters={"Targets": ["<target>"],"ForceUpdate":<force_flag>};type=application/json' --form "UpdateFile=@<image_file>;type=application/octet-stream" </pre>															
Description	<p>This Redfish API shall update the firmware image. The image format must be in fwpkg format, suitable for PLDM. This API can be used when the BMC needs to check for identical or lower versions which are about to be updated.</p>															
Syntax Description	<table border="1"> <tr> <td>user</td> <td>BMC Username</td> </tr> <tr> <td>p/w</td> <td>BMC User password</td> </tr> <tr> <td>bmc_ip</td> <td>BMC IP address</td> </tr> <tr> <td>target</td> <td>Target to be updated: (Only BIOS should be used from BMC) /redfish/v1/UpdateService/FirmwareInventory/MGX_FW_CPU_0</td> </tr> <tr> <td>force_flag</td> <td>Force flag options: true / false When force flag is set to true, the BMC will check if the fwpkg file contains a version number which is lower or identical to the one already installed.</td> </tr> <tr> <td>image_name</td> <td>fwpkg image name and path</td> </tr> <tr> <td>id (return value)</td> <td>task_id_number</td> </tr> </table>	user	BMC Username	p/w	BMC User password	bmc_ip	BMC IP address	target	Target to be updated: (Only BIOS should be used from BMC) /redfish/v1/UpdateService/FirmwareInventory/MGX_FW_CPU_0	force_flag	Force flag options: true / false When force flag is set to true, the BMC will check if the fwpkg file contains a version number which is lower or identical to the one already installed.	image_name	fwpkg image name and path	id (return value)	task_id_number	
user	BMC Username															
p/w	BMC User password															
bmc_ip	BMC IP address															
target	Target to be updated: (Only BIOS should be used from BMC) /redfish/v1/UpdateService/FirmwareInventory/MGX_FW_CPU_0															
force_flag	Force flag options: true / false When force flag is set to true, the BMC will check if the fwpkg file contains a version number which is lower or identical to the one already installed.															
image_name	fwpkg image name and path															
id (return value)	task_id_number															
Default	N/A															
History	88.0002.1040															
Response Example	<pre> { "@odata.id": "/redfish/v1/TaskService/Tasks/0", "@odata.type": "#Task.v1_4_3.Task", "Id": "0", "TaskState": "Running", "TaskStatus": "OK" } </pre>															

Related Commands	Show Firmware Inventory Show Firmware Version & Health Show Update Firmware Status Filter Next Update Firmware End Components
Notes	<ul style="list-style-type: none"> The fwpkg file contains header for PLDM service The header must include information about the desired component to be updated

3.1.7 Show Update Firmware Multipart Status

Redfish API	curl -s -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/TaskService/Tasks/<task_id_number>	
Description	This Redfish API shall be used to get the Update Firmware task state. When task state changes from “Running” to “Completed”, the firmware is fully updated or the firmware update task has a different result.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	task_id_number	Return value taken from Update Firmware command
Default	N/A	
History	88.0002.1040	

Response Example	<p>Response Example for Updating Firmware with Multipart API when using the same formware version as the installed one, with force flag set to 'true':</p> <pre> { "@odata.id": "/redfish/v1/TaskService/Tasks/4", "@odata.type": "#Task.v1_4_3.Task", "EndTime": "2025-04-28T09:05:02+00:00", "HidePayload": false, "Id": "4", "Messages": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The task with Id '4' has started.", "MessageArgs": ["4"], "MessageId": "TaskEvent.1.0.3.TaskStarted", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#MessageRegistry.v1_4_1.MessageRegistry", "Message": "The target device 'MGX_FW_BMC_0' will be updated with image '88.0002.1038'.", "MessageArgs": ["MGX_FW_BMC_0", "88.0002.1038"], "MessageId": "Update.1.0.TargetDetermined", "Resolution": "None.", "Severity": "OK" }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The task with Id '4' has changed to progress 100 percent complete.", "MessageArgs": ["4", "100"], "MessageId": "TaskEvent.1.0.3.TaskProgressChanged", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The task with Id '4' has completed.", "MessageArgs": ["4"], "MessageId": "TaskEvent.1.0.3.TaskCompletedOK", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#MessageRegistry.v1_4_1.MessageRegistry", "Message": "The update operation for the component 'MGX_FW_BMC_0' is skipped because 'Component image is identical'.", "MessageArgs": ["MGX_FW_BMC_0", "Component image is identical"], "MessageId": "NvidiaUpdate.1.0.ComponentUpdateSkipped", "Resolution": "Retry firmware update operation with the force flag", "Severity": "OK" }], "Name": "Task 4", "Payload": { "HttpHeaders": [], "HttpOperation": "POST", "JsonBody": "<discarded>", "TargetUri": "/redfish/v1/UpdateService/update-multipart" }, "PercentComplete": 100, "StartTime": "2025-04-28T09:05:01+00:00", "TaskMonitor": "/redfish/v1/TaskService/Tasks/4/Monitor", "TaskState": "Completed", "TaskStatus": "OK" } </pre>
Related Commands	<p>Show Firmware Inventory Show Firmware Version & Health Updating FW with Multipart API</p>
Notes	

3.1.8 Firmware Update Expected Duration

Component	BIOS
Update time (seconds)	230-260 seconds

3.2 Chassis

The Chassis section is composed of the following APIs:

- [Show Chassis Information](#)
- [Show Chassis Component Information](#)

3.2.1 Show Chassis Information

Redfish API	curl -s -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/Chassis	
Description	This Redfish API shall be used to show all chassis components.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	
Response Example	<pre> { "@odata.id": "/redfish/v1/Chassis", "@odata.type": "#ChassisCollection.ChassisCollection", "Members": [{ "@odata.id": "/redfish/v1/Chassis/BMC_eeeprom" }, { "@odata.id": "/redfish/v1/Chassis/CPLD_0" }, { "@odata.id": "/redfish/v1/Chassis/MGX_BMC_0" }, { "@odata.id": "/redfish/v1/Chassis/MGX_ERoT_BMC_0" }, { "@odata.id": "/redfish/v1/Chassis/MGX_ERoT_CPU_0" }, { "@odata.id": "/redfish/v1/Chassis/MGX_ERoT_FPGA_0" }, { "@odata.id": "/redfish/v1/Chassis/MGX_ERoT_NVSwitch_0" }, { "@odata.id": "/redfish/v1/Chassis/MGX_ERoT_NVSwitch_1" }, { "@odata.id": "/redfish/v1/Chassis/MGX_NVSwitch_0" }, { "@odata.id": "/redfish/v1/Chassis/MGX_NVSwitch_1" }], "Members@odata.count": 10, "Name": "Chassis Collection" } </pre>	
Related Commands		

Notes	
-------	--

3.2.2 Show Chassis Component Information

Redfish API	curl -s -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/Chassis/<chassis_comp>	
Description	This Redfish API shall be used to show a specific chassis component information.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	chassis_comp	Possible Components: CPLD_0, MGX_BMC_0, MGX_ERoT_BMC_0, MGX_ERoT_CPU_0, MGX_ERoT_FPGA_0, MGX_ERoT_NVSwitch_X, MGX_NVSwitch_X
Default	N/A	
History	88.0002.0574	

<p>Response Example</p>	<pre> { "@odata.id": "/redfish/v1/Chassis/MGX_ERoT_CPU_0", "@odata.type": "#Chassis.v1_22_0.Chassis", "Actions": { "Oem": { "#NvidiaRoTChassis.SetIrreversibleConfig": { "@Redfish.ActionInfo": "/redfish/v1/Chassis/MGX_ERoT_CPU_0/Oem/NvidiaRoT/ SetIrreversibleConfigActionInfo", "target": "/redfish/v1/Chassis/MGX_ERoT_CPU_0/Actions/Oem/ NvidiaRoTChassis.SetIrreversibleConfig" } } }, "Certificates": { "@odata.id": "/redfish/v1/Chassis/MGX_ERoT_CPU_0/Certificates" }, "ChassisType": "Component", "Id": "MGX_ERoT_CPU_0", "Links": { "ComputerSystems": [{ "@odata.id": "/redfish/v1/Systems/System_0" }], "ManagedBy": [{ "@odata.id": "/redfish/v1/Managers/BMC_0" }], "Oem": { "Nvidia": { "@odata.type": "#NvidiaChassis.v1_3_0.NvidiaChassis", "ComponentsProtected": [{ "@odata.id": "/redfish/v1/Managers/BMC_0" }] } } }, "Location": { "PartLocation": { "LocationType": "Embedded" } }, "Manufacturer": "NVIDIA", "Name": "MGX_ERoT_CPU_0", "Oem": { "Nvidia": { "@odata.type": "#NvidiaChassis.v1_3_0.NvidiaRoTChassis", "AutomaticBackgroundCopyEnabled": false, "BackgroundCopyStatus": "Pending", "IrreversibleConfigEnabled": false, "RoTProtectedComponents": { "@odata.id": "/redfish/v1/Chassis/MGX_ERoT_CPU_0/Oem/NvidiaRoT/RoTProtectedComponents" } } }, "SKU": "0x4D35368B", "SerialNumber": "0x04120619180A0D28", "Status": { "Conditions": [], "Health": "OK", "HealthRollup": "OK", "State": "Enabled" }, "UUID": "f72d6fc0-5675-11ed-9b6a-0242ac120002" } </pre>
<p>Related Commands</p>	<p>Show Chassis Information</p>
<p>Notes</p>	

3.3 Certificate Service

The Certificate Service section is composed of the following APIs:

- [Show Certificate Service](#)
- [Show Certificate Locations](#)
- [Show Certificate](#)

3.3.1 Show Certificate Service

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X POST https://<bmc_ip>/redfish/v1/CertificateService	
Description	This Redfish API shall be used to view the Certificate Service used for https connections.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	
Response Example	<pre> { "@odata.id": "/redfish/v1/CertificateService", "@odata.type": "#CertificateService.v1_0_0.CertificateService", "Actions": { "#CertificateService.GenerateCSR": { "target": "/redfish/v1/CertificateService/Actions/ CertificateService.GenerateCSR" }, "#CertificateService.ReplaceCertificate": { "CertificateType@Redfish.AllowableValues": ["PEM"], "target": "/redfish/v1/CertificateService/Actions/ CertificateService.ReplaceCertificate" } }, "CertificateLocations": { "@odata.id": "/redfish/v1/CertificateService/CertificateLocations" }, "Description": "Actions available to manage certificates", "Id": "CertificateService", "Name": "Certificate Service" } </pre>	

3.3.2 Show Certificate Locations

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X POST https://<bmc_ip>/redfish/v1/CertificateService/CertificateLocations	
Description	This Redfish API shall be used to show the https certificate location.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	

Response Example	<pre>{ "@odata.id": "/redfish/v1/CertificateService/CertificateLocations", "@odata.type": "#CertificateLocations.v1_0_0.CertificateLocations", "Description": "Defines a resource that an administrator can use in order to locate all certificates installed on a given service", "Id": "CertificateLocations", "Links": { "Certificates": [{ "@odata.id": "/redfish/v1/Managers/BMC_0/NetworkProtocol/HTTPS/Certificates/1" }] }, "Certificates@odata.count": 1 }, "Name": "Certificate Locations" }</pre>
Related Commands	Show Certificate Service

3.3.3 Show Certificate

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X POST https://<bmc_ip>/redfish/v1/Managers/BMC_0/NetworkProtocol/HTTPS/Certificates/1	
Description	This Redfish API shall be used to show the https certificate location.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	
Response Example	<pre>{ "@odata.id": "/redfish/v1/Managers/BMC_0/NetworkProtocol/HTTPS/Certificates/1", "@odata.type": "#Certificate.v1_0_0.Certificate", "CertificateString": "-----BEGIN CERTIFICATE-----\nMIICODCCAbGgAwIBAgIEYHm/ QDAKBggqhk*****-----END CERTIFICATE-----\n", "Description": "HTTPS Certificate", "Id": "1", "Issuer": { "CommonName": "testhost", "Country": "US", "Organization": "OpenBMC" }, "KeyUsage": ["KeyEncipherment", "ServerAuthentication"], "Name": "HTTPS Certificate", "Subject": { "CommonName": "testhost", "Country": "US", "Organization": "OpenBMC" }, "ValidNotAfter": "2035-01-25T08:14:02+00:00", "ValidNotBefore": "2025-01-27T08:14:02+00:00" }</pre>	
Related Commands	Show Certificate Service, Show Certificate Locations	
Notes		

3.4 Session Service

The Session Service section is composed of the following APIs:

- [Show Session Service](#)
- [Show Sessions](#)
- [Show Session Details](#)

3.4.1 Show Session Service

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X POST https://<bmc_ip>/redfish/v1/SessionService	
Description	This Redfish API shall be used to view Session Service properties	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	
Response Example	<pre>{ "@odata.id": "/redfish/v1/SessionService", "@odata.type": "#SessionService.v1_0_2.SessionService", "Description": "Session Service", "Id": "SessionService", "Name": "Session Service", "ServiceEnabled": true, "SessionTimeout": 1800, "Sessions": { "@odata.id": "/redfish/v1/SessionService/Sessions" } }</pre>	
Related Commands		
Notes		

3.4.2 Show Sessions

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X POST https://<bmc_ip>/redfish/v1/SessionService/Sessions	
Description	This Redfish API shall be used to show the open sessions on the Session service.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	

Response Example	<pre>{ "@odata.id": "/redfish/v1/SessionService/Sessions", "@odata.type": "#SessionCollection.SessionCollection", "Description": "Session Collection", "Members": [{ "@odata.id": "/redfish/v1/SessionService/Sessions/atfDOAJ59o" }, { "@odata.id": "/redfish/v1/SessionService/Sessions/8qcRf9Puda" }], "Members@odata.count": 2, "Name": "Session Collection" }</pre>
Related Commands	Show Session Service
Notes	

3.4.3 Show Session Details

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X POST https://<bmc_ip>/redfish/v1/SessionService/Sessions/<session_id>	
Description	This Redfish API shall be used to show the open sessions on the Session service.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	session_id	This ID value should be taken from the Show Sessions RF API.
Default	N/A	
History	88.0002.0574	
Response Example	<pre>{ "@odata.id": "/redfish/v1/SessionService/Sessions/8qcRf9Puda", "@odata.type": "#Session.v1_7_0.Session", "ClientOriginIPAddress": "10.0.1.2", "Description": "Manager User Session", "Id": "8qcRf9Puda", "Name": "User Session", "Roles": ["Administrator"], "UserName": "admin" }</pre>	
Related Commands	Show Session Service, Show Sessions	
Notes		

3.5 User Management

The User Management section is composed of several APIs which work in conjunction:

- [Show BMC User Account Configuration](#)
- [Show BMC User Accounts](#)
- [Create New BMC User](#)

- [Change BMC User Password](#)
- [Change BMC "root" User Account Password from BMC "admin" User Account](#)
- [Change BMC User Account Permissions](#)
- [Change BMC Account Lockout Duration](#)
- [Change BMC Minimum Password Length](#)
- [Change BMC Account Lockout Threshold](#)
- [Show BMC Account Service Roles](#)
- [Show BMC User Account Service Role Options](#)



The password locking policy states that after 10 consecutive unsuccessful login attempts, following a factory reset, the user account will be locked for a duration of 20 seconds.

Minimum p/w length following a factory reset is 12 characters.

3.5.1 Show BMC User Account Configuration

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/AccountService	
Description	This Redfish API shall list all applicable BMC user accounts.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	

Response Example	<pre> { "@odata.id": "/redfish/v1/AccountService", "@odata.type": "#AccountService.v1_15_0.AccountService", "AccountLockoutDuration": 20, "AccountLockoutThreshold": 10, "Accounts": { "@odata.id": "/redfish/v1/AccountService/Accounts" }, "Description": "Account Service", "HTTPBasicAuth": "Enabled", "HTTPBasicAuth@AllowableValues": ["Enabled", "Disabled"], "Id": "AccountService", "LDAP": { "Certificates": { "@odata.id": "/redfish/v1/AccountService/LDAP/Certificates" } }, "MaxPasswordLength": 20, "MinPasswordLength": 12, "MultiFactorAuth": { "ClientCertificate": { "CertificateMappingAttribute": "CommonName", "Certificates": { "@odata.id": "/redfish/v1/AccountService/MultiFactorAuth/ClientCertificate/ Certificates", "@odata.type": "#CertificateCollection.CertificateCollection", "Members": [], "Members@odata.count": 0 }, "Enabled": true, "RespondToUnauthenticatedClients": true } }, "Name": "Account Service", "Oem": { "OpenBMC": { "@odata.id": "/redfish/v1/AccountService#/Oem/OpenBMC", "@odata.type": "#OpenBMCAccountService.v1_0_0.AccountService", "AuthMethods": { "BasicAuth": true, "Cookie": true, "SessionToken": true, "TLS": true, "XToken": true } } }, "Roles": { "@odata.id": "/redfish/v1/AccountService/Roles" }, "ServiceEnabled": true } </pre>
Related Commands	All User Account APIs
Notes	

3.5.2 Show BMC User Accounts

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/AccountService/Accounts	
Description	This Redfish API shall list all applicable BMC user accounts.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	
Response Example	<pre>{ "@odata.id": "/redfish/v1/AccountService/Accounts", "@odata.type": "#ManagerAccountCollection.ManagerAccountCollection", "Description": "BMC User Accounts", "Members": [{ "@odata.id": "/redfish/v1/AccountService/Accounts/admin" }, { "@odata.id": "/redfish/v1/AccountService/Accounts/root" }], "Members@odata.count": 2, "Name": "Accounts Collection" }</pre>	
Related Commands	Change BMC User Password	
Notes		

3.5.3 Create New BMC User

Redfish API	curl -k -u <user>:<p/w> -X POST https://<bmc_ip>/redfish/v1/AccountService/Accounts -d '{ "UserName": "<new_user_name>", "Password": "new_user_p/w", "RoleId": "<role_id>", "Enabled": true }'	
Description	This Redfish API shall list all applicable BMC user accounts.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	new_user_name	The new BMC user's user name
	new_user_p/w	The new BMC user's password
	role_id	BMC Role ID options: Administrator, Operator, ReadOnly <ul style="list-style-type: none"> Administrator: Users are allowed to configure all OpenBMC (including user-management, network and all configurations). Users will have full administrative access. Operator: Users are allowed to view and control basic operations. This includes reboot of the host, etc. But users are not allowed to change other configuration like user, network, and so forth. ReadOnly: Users only have read access and cannot change any behavior of the system.

Default	N/A
History	88.0002.0956
Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The resource was created successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Created", "MessageSeverity": "OK", "Resolution": "None." }] } </pre>
Related Commands	Change BMC User Password Show BMC User Accounts
Notes	

3.5.4 Change BMC User Password

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X PATCH -d '{"Password": "<new_password>"}' https://<bmc_ip>/redfish/v1/AccountService/Accounts/<user_to_be_patched>	
Description	This Redfish API for user management interface may be used to configure the new password or to alter the default password.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	user_to_be_patched	BMC user to be updated
	new_password	BMC new password (e.g., "switch_bmc@Nvid1a")
Default	N/A	
History	88.0002.0574	
Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.16.0.Success", "MessageSeverity": "OK", "Resolution": "None" }] } </pre>	
Related Commands	Show BMC Users	
Notes	<p>The BMC password must comply with the following policy parameters:</p> <ul style="list-style-type: none"> Using ASCII and Unicode characters is permitted Minimum length: 12 (applies after factory reset) <p>The following is a valid example password: HelloNvidia3D!</p>	

3.5.5 Change BMC "root" User Account Password from BMC "admin" User Account

Redfish API	curl -k -u admin_user:<admin p/w> -X PATCH -d '{"Password":"new_root_p/w"}' https://<bmc_ip>/redfish/v1/AccountService/Accounts/root	
Description	This Redfish API for user management interface may be used to configure the new password or to alter the default password.	
Syntax Description	admin_user	BMC Username admin
	admin p/w	BMC User password admin
	bmc_ip	BMC IP address
	new root p/w	BMC new password for User root (e.g., "juliet_bmc@Nvid1a")
Default	N/A	
History	88.0002.0574	
Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.16.0.Success", "MessageSeverity": "OK", "Resolution": "None" }] } </pre>	
Related Commands	Change BMC Users	
Notes	<p>The BMC password must comply with the following policy parameters:</p> <ul style="list-style-type: none"> Using ASCII and Unicode characters is permitted Minimum length: 12 <p>The following is a valid example password: HelloNvidia3D!</p>	

3.5.6 Change BMC User Account Permissions

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X PATCH -d '{"RoleId": "<new_role_id>" }' https://<bmc_ip>/redfish/v1/AccountService/Accounts/<user_to_patched>	
Description	This Redfish API for user management interface may be used to configure the new password or to alter the default password.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	user_to_be_patched	BMC user to be updated

	new_role_id	The new user's role ID (Can be either: Administrator, Operator, ReadOnly) <ul style="list-style-type: none"> Administrator: Users are allowed to configure all OpenBMC (including user-management, network and all configurations). Users will have full administrative access. Operator: Users are allowed to view and control basic operations. This includes reboot of the host, etc. But users are not allowed to change other configuration like user, network, and so forth. ReadOnly: Users only have read access and cannot change any behavior of the system.
Default	N/A	
History	88.0002.0956	
Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.16.0.Success", "MessageSeverity": "OK", "Resolution": "None" }] } </pre>	
Related Commands	Show BMC Users	
Notes	<p>The BMC password must comply with the following policy parameters:</p> <ul style="list-style-type: none"> Using ASCII and Unicode characters is permitted Minimum length: 12 (applies after factory reset) <p>The following is a valid example password: HelloNvidia3D!</p>	

3.5.7 Change BMC Account Lockout Duration

Redfish API	curl -k -u <user>:<p/w> -X PATCH https://<bmc_ip>/redfish/v1/AccountService -d '{"AccountLockoutDuration":<duration>}'	
Description	This Redfish API shall be used to change the BMC account lockout duration.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	duration	Lockout duration time in seconds (should be greater than 20)
Default	N/A	
History	88.0002.0956	

Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }] } </pre>
Related Commands	Show BMC User Account Configuration
Notes	

3.5.8 Change BMC Minimum Password Length

Redfish API	curl -k -u <user>:<p/w> -X PATCH https://<bmc_ip>/redfish/v1/AccountService -d '{"MinPasswordLength":<length>}'	
Description	This Redfish API shall be used to change the BMC account lockout duration.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	length	Number of characters (should be greater than 12 and lower than 256)
Default	N/A	
History	88.0002.0956	
Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }] } </pre>	
Related Commands	Show BMC User Account Configuration	
Notes		

3.5.9 Change BMC Account Lockout Threshold

Redfish API	curl -k -u <user>:<p/w> -X PATCH https://<bmc_ip>/redfish/v1/AccountService -d '{"AccountLockoutThreshold":<threshold>}'	
Description	This Redfish API shall be used to change the BMC account lockout duration.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	threshold	Lockout threshold counts (should be greater than 10).
Default	N/A	
History	88.0002.0956	
Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }] } </pre>	
Related Commands	Show BMC User Account Configuration	
Notes		

3.5.10 Show BMC Account Service Roles

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/AccountService/Roles	
Description	This Redfish API shall list all applicable BMC user accounts.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	

Response Example	<pre> { "@odata.id": "/redfish/v1/AccountService/Roles", "@odata.type": "#RoleCollection.RoleCollection", "Description": "BMC User Roles", "Members": [{ "@odata.id": "/redfish/v1/AccountService/Roles/Administrator" }, { "@odata.id": "/redfish/v1/AccountService/Roles/Operator" }, { "@odata.id": "/redfish/v1/AccountService/Roles/ReadOnly" }], "Members@odata.count": 3, "Name": "Roles Collection" } </pre>
Related Commands	Show BMC Users Accounts
Notes	

3.5.11 Show BMC User Account Service Role Options

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/AccountService/Roles/<role>	
Description	This Redfish API shall list all applicable BMC user accounts.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	role	Administrator, Operator, ReadOnly <ul style="list-style-type: none"> Administrator: Users are allowed to configure all OpenBMC (including user-management, network and all configurations). Users will have full administrative access. Operator: Users are allowed to view and control basic operations. This includes reboot of the host, etc. But users are not allowed to change other configuration like user, network, and so forth. ReadOnly: Users only have read access and cannot change any behavior of the system.
Default	N/A	
History	88.0002.0574	

Response Example	<pre> { "@odata.id": "/redfish/v1/AccountService/Accounts/root", "@odata.type": "#ManagerAccount.v1_7_0.ManagerAccount", "AccountTypes": ["HostConsole", "IPMI", "Redfish", "WebUI", "ManagerConsole"], "Description": "User Account", "Enabled": true, "Id": "root", "Links": { "Role": { "@odata.id": "/redfish/v1/AccountService/Roles/Administrator" } }, "Locked": false, "Locked@Redfish.AllowableValues": ["false"], "Name": "User Account", "Password": null, "PasswordChangeRequired": false, "RoleId": "Administrator", "StrictAccountTypes": true, "UserName": "root" } </pre>
Related Commands	Show BMC User Account Service Roles
Notes	

3.6 Attestation

The Attestation section is composed of the following APIs, some of which work in conjunction:

- [Show System EROT List](#)
- [Show EROT Security Information](#)
- [Generate EROT SPDM Information](#)
- [Show EROT SPDM Generation Status](#)
- [Show EROT SPDM Information](#)
- [Set EROT Automatic Background Copy State](#)
- [Show EROT Automatic Background Copy Status](#)
- [Show Minimum Security Version Information for Application Firmware](#)

Flow for generating EROT SPDM information:

1. Run [Generate EROT SPDM Information](#) API initiates the task of generating EROT SPDM information in the system
2. Run [Show EROT SPDM Generation Status](#) API to figure out that task state (i.e., whether a task is still running or completed).
3. Pull on step 2 until task completion.
4. Run [Show EROT SPDM Information](#) API to see that relevant EROT SPDM information.

3.6.1 Show System EROT List

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/ComponentIntegrity/
Description	This Redfish API shall be used to collect information about the system's EROT (BMC, CPU, FPGA, NVSwitch0, NVSwitch1).

Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	
Response Example	<pre> { "@odata.id": "/redfish/v1/ComponentIntegrity", "@odata.type": "#ComponentIntegrityCollection.ComponentIntegrityCollection", "Members": [{ "@odata.id": "/redfish/v1/ComponentIntegrity/MGX_ERoT_BMC_0" }, { "@odata.id": "/redfish/v1/ComponentIntegrity/MGX_ERoT_CPU_0" }, { "@odata.id": "/redfish/v1/ComponentIntegrity/MGX_ERoT_FPGA_0" }, { "@odata.id": "/redfish/v1/ComponentIntegrity/MGX_ERoT_NVSwitch_0" }, { "@odata.id": "/redfish/v1/ComponentIntegrity/MGX_ERoT_NVSwitch_1" }], "Members@odata.count": 5, "Name": "ComponentIntegrity Collection" } </pre>	
Related Commands	Show ERoT Security Information Generate ERoT SPDm Information Show ERoT SPDm Generation Status Show ERoT SPDm Information Set ERoT Automatic Background Copy State Show ERoT Automatic Background Copy State	
Notes		

3.6.2 Show ERoT Security Information

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/ComponentIntegrity/<comp_name>	
Description	This Redfish API shall be used to provide critical and pertinent security information about a specific ERoT device.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	comp_name	MGX_ERoT_BMC_0, MGX_ERoT_FPGA_0, etc.
Default	N/A	
History	88.0002.0574	

Response Example	<pre> { "@odata.id": "/redfish/v1/ComponentIntegrity/MGX_ERoT_BMC_0", "@odata.type": "#ComponentIntegrity.v1_0_0.ComponentIntegrity", "Actions": { "#ComponentIntegrity.SPDMGetSignedMeasurements": { "@Redfish.ActionInfo": "/redfish/v1/ComponentIntegrity/MGX_ERoT_BMC_0/SPDMGetSignedMeasurementsActionInfo", "target": "/redfish/v1/ComponentIntegrity/MGX_ERoT_BMC_0/Actions/ComponentIntegrity.SPDMGetSignedMeasurements" } }, "ComponentIntegrityEnabled": true, "ComponentIntegrityType": "SPDM", "ComponentIntegrityTypeVersion": "unknown", "Id": "MGX_ERoT_BMC_0", "Links": { "ComponentsProtected": [{ "@odata.id": "/redfish/v1/Managers/BMC_0" }] }, "Name": "SPDM Integrity for MGX_ERoT_BMC_0", "SPDM": { "IdentityAuthentication": { "ResponderAuthentication": { "ComponentCertificate": { "@odata.id": "/redfish/v1/Chassis/MGX_ERoT_BMC_0/Certificates/CertChain" } } }, "Requester": { "@odata.id": "/redfish/v1/Managers/BMC_0" } }, "TargetComponentURI": "/redfish/v1/Chassis/MGX_ERoT_BMC_0" } </pre>
Related Commands	<p>Show System EROT List Generate EROT SPDM Information Show EROT SPDM Generation Status Show EROT SPDM Information Set EROT Automatic Background Copy State Show EROT Automatic Background Copy State</p>
Notes	

3.6.3 Generate EROT SPDM Information

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X POST https://<bmc_ip>/redfish/v1/ComponentIntegrity/<comp_name>/Actions/ComponentIntegrity.SPDMGetSignedMeasurements	
Description	This Redfish API shall be used to generate an SPDM cryptographic signed statement over the given nonce and measurements of the SPDM Responder.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	comp_name	MGX_ERoT_BMC_0, MGX_ERoT_FPGA_0, etc.
	Id (return value)	task_id_number
Default	N/A	
History	88.0002.0574	

Response Example	<pre>{ "@odata.id": "/redfish/v1/TaskService/Tasks/0", "@odata.type": "#Task.v1_4_3.Task", "Id": "0", "TaskState": "Running", "TaskStatus": "OK" }</pre>
Related Commands	<p>Show System EROT List Show EROT Security Information Show EROT SPDM Generation Status Show EROT SPDM Information Set EROT Automatic Background Copy State Show EROT Automatic Background Copy State</p>
Notes	

3.6.4 Show EROT SPDM Generation Status

Redfish API	curl -s -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/TaskService/Tasks/<task_id_number>	
Description	This Redfish API shall be used to get the EROT SPDM Generation task state. When task state changes from "Running" to "Completed", the SPDM information is ready to be displayed.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	task_id_number	Return value taken from Generate EROT SPDM Information command
Default	N/A	
History	88.0002.0574	

Response Example	<pre> { "@odata.id": "/redfish/v1/TaskService/Tasks/0", "@odata.type": "#Task.v1_4_3.Task", "EndTime": "2024-06-25T05:13:08+00:00", "HidePayload": false, "Id": "0", "Messages": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The task with Id '0' has started.", "MessageArgs": ["0"], "MessageId": "TaskEvent.1.0.3.TaskStarted", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The task with Id '0' has completed.", "MessageArgs": ["0"], "MessageId": "TaskEvent.1.0.3.TaskCompletedOK", "MessageSeverity": "OK", "Resolution": "None." }], "Name": "Task 0", "Payload": { "HttpHeaders": [{ "Location": "/redfish/v1/ComponentIntegrity/MGX_ERoT_BMC_0/Actions/ComponentIntegrity.SPDMGetSignedMeasurements/data" }], "HttpOperation": "POST", "JsonBody": "<discarded>", "TargetUri": "/redfish/v1/ComponentIntegrity/MGX_ERoT_BMC_0/Actions/ComponentIntegrity.SPDMGetSignedMeasurements" }, "PercentComplete": 100, "StartTime": "2024-06-25T05:13:07+00:00", "TaskMonitor": "/redfish/v1/TaskService/Tasks/0/Monitor", "TaskState": "Completed", "TaskStatus": "OK" } </pre>
Related Commands	<p> Show System ERoT List Show ERoT Security Information Generate ERoT SPDM Information Show ERoT SPDM Information Set ERoT Automatic Background Copy State Show ERoT Automatic Background Copy State </p>
Notes	

3.6.5 Show ERoT SPDM Information

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/ComponentIntegrity/<comp_name>/Actions/ComponentIntegrity.SPDMGetSignedMeasurements/data	
Description	Use the API below to collect the Signed Measurement data.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	comp_name	MGX_ERoT_BMC_0, MGX_ERoT_FPGA_0, and so forth
Default	N/A	
History	88.0002.0574	

Response Example	<pre>{ "HashingAlgorithm": "TPM_ALG_SHA_384", "SignedMeasurements": "EeAB/ 6LLdRTg+iEumJY8eeaRRSMahEHw26QQNaVqerI6Sx+RABFgAABACQkAAQEHAIMEAAIAAAA**Rest of signature is left out***", "SigningAlgorithm": "TPM_ALG_ECDSA_ECC_NIST_P384", "Version": "1.1.0" }</pre>
Related Commands	<p>Show System ERoT List Show ERoT Security Information Generate ERoT SPDm Information Show ERoT SPDm Generation Status Set ERoT Automatic Background Copy State Show ERoT Automatic Background Copy State</p>
Notes	

3.6.6 Set ERoT Automatic Background Copy State

Redfish API	<pre>curl -k -u <user>:<p/w> -X PATCH -d '{"Oem": {"Nvidia": {"AutomaticBackgroundCopyEnabled": <state>}}}' https://<bmc_ip>/redfish/v1/Chassis/ <ERoT Name></pre>	
Description	This Redfish API shall be used to set the ERoT Automatic Background Copy State.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	state	True means that firmware update through this ERoT will be non blocking, False means it will be blocking.
	bmc_ip	BMC IP address
	ERoT Name	Name of the ERoT
Default	N/A	
History	88.0002.0574	
Response Example	<pre>{ "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.16.0.Success", "MessageSeverity": "OK", "Resolution": "None" }] }</pre>	
Related Commands	<p>Show System ERoT List Show ERoT Security Information Generate ERoT SPDm Information Show ERoT SPDm Generation Status Show ERoT SPDm Information Show ERoT Automatic Background Copy State</p>	
Notes		

3.6.7 Show EROT Automatic Background Copy State

Redfish API	curl -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/Chassis/<ERoT Name>	
Description	This Redfish API shall be used to show the EROT Automatic Background Copy State.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	ERoT Name	Name of the ERoT
Default	N/A	
History	88.0002.0574	
Response Example	<pre> { "@odata.id": "/redfish/v1/Chassis/MGX_ERoT_NVSwitch_0", "@odata.type": "#Chassis.v1_17_0.Chassis", "Actions": { "Oem": { "#CAKInstall": { "target": "/redfish/v1/Chassis/MGX_ERoT_NVSwitch_0/Actions/Oem/CAKInstall" }, "#CAKLock": { "target": "/redfish/v1/Chassis/MGX_ERoT_NVSwitch_0/Actions/Oem/CAKLock" }, "#CAKTest": { "target": "/redfish/v1/Chassis/MGX_ERoT_NVSwitch_0/Actions/Oem/CAKTest" }, "#DOTDisable": { "target": "/redfish/v1/Chassis/MGX_ERoT_NVSwitch_0/Actions/Oem/DOTDisable" }, "#DOTTokenInstall": { "target": "/redfish/v1/Chassis/MGX_ERoT_NVSwitch_0/Actions/Oem/DOTTokenInstall" }, "Nvidia": { "#BootProtectedDevice": { "target": "/redfish/v1/Chassis/MGX_ERoT_NVSwitch_0/Actions/Oem/Nvidia/BootProtectedDevice" } } } }, <.....Snip.....> "Manufacturer": "NVIDIA", "Name": "MGX_ERoT_NVSwitch_0", "Oem": { "Nvidia": { "@odata.type": "#NvidiaChassis.v1_1_0.NvidiaChassis", "AutomaticBackgroundCopyEnabled": false, "BackgroundCopyStatus": "Completed", "InbandUpdatePolicyEnabled": true, "ManualBootModeEnabled": false } }, "SKU": "0x4D35368B", "SerialNumber": "0x011E031A1810250A", "Status": { "Conditions": [], "Health": "OK", "HealthRollup": "OK", "State": "Enabled" }, "UUID": "f72d6fb0-5675-11ed-9b6a-0242ac120002" } </pre>	
Related Commands	Show System EROT List Show EROT Security Information Generate EROT SPDM Information Show EROT SPDM Generation Status Show EROT SPDM Information Set EROT Automatic Background Copy State	
Notes		

3.6.8 Show Minimum Security Version Information for Application Firmware

Redfish API	curl -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/Chassis/<ERoT Name>/Oem/NvidiaRoT/RoTProtectedComponents/<AP Name>	
Description	This Redfish API shall be used to display the minimum security version information for the application firmware. The value of "MinimumSecurityVersion" is the lowest possible security version which can be installed.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	ERoT Name (paired with AP Name)	Name of the ERoT. Possible values: MGX_ERoT_BMC_0, MGX_ERoT_CPU_0, MGX_ERoT_FPGA_0, MGX_ERoT_NVSwitch_0, MGX_ERoT_NVSwitch_1
	AP Name (paired with ERoT Name)	Name of Applicaton. Possible values: MGX_BMC_0, MGX_CPU_0, MGX_FPGA_0, MGX_NVSwitch_0, MGX_NVSwitch_1
Default	N/A	
History	88.0002.1040	

Response Example

```
{
  "@Redfish.Settings": {
    "@odata.type": "#Settings.v1_3_3.Settings",
    "SettingsObject": {
      "@odata.id": "/redfish/v1/Chassis/MGX_ERoT_BMC_0/Oem/NvidiaRoT/
RoTProtectedComponents/MGX_BMC_0/Settings"
    }
  },
  "@odata.id": "/redfish/v1/Chassis/MGX_ERoT_BMC_0/Oem/NvidiaRoT/RoTProtectedComponents/
MGX_BMC_0",
  "@odata.type": "#NvidiaRoTProtectedComponent.v1_0_0.NvidiaRoTProtectedComponent",
  "Actions": {
    "#NvidiaRoTProtectedComponent.RevokeKeys": {
      "@Redfish.ActionInfo": "/redfish/v1/Chassis/MGX_ERoT_BMC_0/Oem/NvidiaRoT/
RoTProtectedComponents/MGX_BMC_0/RevokeKeysActionInfo",
      "target": "/redfish/v1/Chassis/MGX_ERoT_BMC_0/Oem/NvidiaRoT/RoTProtectedComponents/
MGX_BMC_0/Actions/NvidiaRoTProtectedComponent.RevokeKeys"
    },
    "#NvidiaRoTProtectedComponent.UpdateMinimumSecurityVersion": {
      "@Redfish.ActionInfo": "/redfish/v1/Chassis/MGX_ERoT_BMC_0/Oem/NvidiaRoT/
RoTProtectedComponents/MGX_BMC_0/UpdateMinimumSecurityVersionActionInfo",
      "target": "/redfish/v1/Chassis/MGX_ERoT_BMC_0/Oem/NvidiaRoT/RoTProtectedComponents/
MGX_BMC_0/Actions/NvidiaRoTProtectedComponent.UpdateMinimumSecurityVersion"
    }
  },
  "ActiveKeySetIdentifier": 25,
  "AllowedKeyIndices": [
    25,
    26,
    27,
    28,
    29,
    30,
    31
  ],
  "BootStatusCode": "0x000601fd00501120",
  "Id": "MGX_BMC_0",
  "ImageSlots": {
    "@odata.id": "/redfish/v1/Chassis/MGX_ERoT_BMC_0/Oem/NvidiaRoT/
RoTProtectedComponents/MGX_BMC_0/ImageSlots"
  },
  "MinimumSecurityVersion": 1,
  "Name": "MGX_ERoT_BMC_0 RoTProtectedComponent MGX_BMC_0",
  "RevokedKeyIndices": [
    0,
    1,
    2,
    3,
    4,
    5,
    6,
    7,
    8,
    9,
    10,
    11,
    12,
    13,
    14,
    15,
    16,
    17,
    18,
    19,
    20,
    21,
    22,
    23,
    24,
    32,
    33,
    34,
    35,
    36,
    37,
    38,
    39,
    40,
    41,
    42,
    43,
    44,
    45,
    46,
    47,
    48,
    49,
    50,
    51,
    52,
    53,
    54,
    55,
    56,
    57,
    58,
    59,
    60,
  ]
}
```

	<pre> 61, 62, 63], "RoTProtectedComponentType": "AP" } </pre>
Related Commands	Show System EROT List Show EROT Security Information Generate EROT SPDM Information Show EROT SPDM Generation Status Show EROT SPDM Information Set EROT Automatic Background Copy State
Notes	

3.7 Power Management

The Power Management section is composed of the following APIs:

- [Apply CPU Host Reset](#)
- [CPU Host Reset Action Information](#)
- [Apply BMC Manager Reset](#)
- [BMC Host Reset Action Information](#)

3.7.1 Apply CPU Host Reset

Redfish API	<pre> curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X POST https:// <bmc_ip>/redfish/v1/Systems/System_0/Actions/ComputerSystem.Reset -d {'ResetType': "<reset_type>"} </pre>	
Description	This Redfish API shall be used to reset the COMe CPU and ASIC subsystems or the whole unit.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	reset_type	<ul style="list-style-type: none"> • GracefulRestart: Gracefully Reset the CPU and ASICs only in case of failure. • ForceRestart: Reset the CPU and ASICs only only in case of failure. • PowerCycle: Power cycle the system. • ForceOn: Power on the Host CPU and ASICs • On: Power on the Host CPU and ASICs • ForceOff: power off immediately the Host CPU and ASICs • GracefulShutdown: Gracefully power off immediately the Host CPU and ASICs
Default	N/A	
History	88.0002.0574 88.0002.1040: Added reset types	

Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }] } </pre>
Related Commands	
Notes	

3.7.2 CPU Host Reset Action Information

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/Systems/System_0/ResetActionInfo	
Description	This Redfish API shall be used to display the CPU Host Reset Action information.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.1040	
Response Example	<pre> { "@odata.id": "/redfish/v1/Systems/System_0/ResetActionInfo", "@odata.type": "#ActionInfo.v1_1_2.ActionInfo", "Id": "ResetActionInfo", "Name": "Reset Action Info", "Parameters": [{ "AllowableValues": ["ForceOff", "PowerCycle", "GracefulShutdown", "On", "ForceOn", "GracefulRestart", "ForceRestart"], "DataType": "String", "Name": "ResetType", "Required": true }] } </pre>	
Related Commands	Apply CPU Host Reset	
Notes		

3.7.3 Apply BMC Manager Reset

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X POST https://<bmc_ip>/redfish/v1/Managers/BMC_0/Actions/Manager.Reset -d '{"ResetType": "<reset_type>"}'	
Description	This Redfish API shall be used to reset the BMC.	
Syntax Description	user	BMC Username

	p/w	BMC User password
	bmc_ip	BMC IP address
	reset_type	<ul style="list-style-type: none"> GracefulRestart: Gracefully Reset the BMC ForceRestart: Force Reset the BMC in case of failure GracefulShutdown: Gracefully power off the BMC
Default	N/A	
History	88.0002.1040	
Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }] } </pre>	
Related Commands		
Notes		

3.7.4 BMC Host Reset Action Information

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/Managers/BMC_0/ResetActionInfo	
Description	This Redfish API shall be used to display the BMC Reset Action information.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.1040	
Response Example	<pre> { "@odata.id": "/redfish/v1/Managers/BMC_0/ResetActionInfo", "@odata.type": "#ActionInfo.v1_1_2.ActionInfo", "Id": "ResetActionInfo", "Name": "Reset Action Info", "Parameters": [{ "AllowableValues": ["GracefulRestart", "ForceRestart", "GracefulShutdown"], "DataType": "String", "Name": "ResetType", "Required": true }] } </pre>	
Related Commands	Apply BMC Reset	
Notes		

3.8 Time Management

Time management can be handled with the following APIs:

- [Show NTP Servers Status](#)
- [Enable/Disable NTP Server](#)
- [Show Time and Date](#)
- [Set Time and Date Manually](#)

3.8.1 Show NTP Servers Status

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/Managers/BMC_0/NetworkProtocol	
Description	This Redfish API shall be used to show the NTP Time Servers status on the BMC.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.1040	
Response Example	<pre> { "@odata.id": "/redfish/v1/Managers/BMC_0/NetworkProtocol", "@odata.type": "#ManagerNetworkProtocol.v1_5_0.ManagerNetworkProtocol", "Description": "Manager Network Service", "FQDN": "juliet-bmc", "HTTP": { "Port": null, "ProtocolEnabled": false }, "HTTPS": { "Certificates": { "@odata.id": "/redfish/v1/Managers/BMC_0/NetworkProtocol/HTTPS/Certificates" }, "Port": 443, "ProtocolEnabled": true }, "HostName": "juliet-bmc", "Id": "NetworkProtocol", "NTP": { "NTPServers": [], "ProtocolEnabled": true }, "Name": "Manager Network Protocol", "Oem": { "Nvidia": { "@odata.type": "#NvidiaNetworkProtocol.v1_0_0.NvidiaNetworkProtocol", "Rsyslog": { "Address": "", "Filter": { "Facilities": [], "LowestSeverity": "Error" }, "Port": 0, "State": "Disabled", "TLS": "Disabled", "TransportProtocol": "TCP" } } }, "SSH": { "Port": 22, "ProtocolEnabled": true }, "Status": { "Health": "OK", "HealthRollup": "OK", "State": "Enabled" } } </pre>	
Related Commands	Enable/Disable NTP Servers	

Notes	
-------	--

3.8.2 Enable/Disable NTP Servers

Redfish API	curl -k -u <user>:<p/w> https://<bmc_ip>/redfish/v1/Managers/BMC_0/NetworkProtocol -H content-type:application/json -X PATCH -d '{"NTP": {"ProtocolEnabled": ntp_value}}'	
Description	This Redfish API shall be used to enable or disable NTP Time Servers on the BMC.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	ntp_value	NTP Value: possible values: true, false.
Default	N/A	
History	88.0002.1040	
Response Example	<input type="text"/>	
Related Commands	Show NTP Servers Status Set Time and Date Manually	
Notes		

3.8.3 Show Time and Date

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/Managers/BMC_0	
Description	This Redfish API shall be used to show the Date and Time on the BMC.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.1040	

Response Example

```
{
  "@odata.id": "/redfish/v1/Managers/BMC_0",
  "@odata.type": "#Manager.v1_15_0.Manager",
  "Actions": {
    "#Manager.Reset": {
      "@Redfish.ActionInfo": "/redfish/v1/Managers/BMC_0/ResetActionInfo",
      "target": "/redfish/v1/Managers/BMC_0/Actions/Manager.Reset"
    },
    "#Manager.ResetToDefaults": {
      "ResetType@Redfish.AllowableValues": [
        "ResetAll"
      ],
      "target": "/redfish/v1/Managers/BMC_0/Actions/Manager.ResetToDefaults"
    },
    "Oem": {
      "#NvidiaManager.AsyncOOBRawCommand": {
        "@Redfish.ActionInfo": "/redfish/v1/Managers/BMC_0/Oem/Nvidia/AsyncOOBRawCommandActionInfo",
        "target": "/redfish/v1/Managers/BMC_0/Actions/Oem/NvidiaManager.AsyncOOBRawCommand"
      },
      "#NvidiaManager.ResetToDefaults": {
        "target": "/redfish/v1/Managers/BMC_0/Actions/Oem/NvidiaManager.ResetToDefaults"
      },
      "#NvidiaManager.SyncOOBRawCommand": {
        "@Redfish.ActionInfo": "/redfish/v1/Managers/BMC_0/Oem/Nvidia/SyncOOBRawCommandActionInfo",
        "target": "/redfish/v1/Managers/BMC_0/Actions/Oem/NvidiaManager.SyncOOBRawCommand"
      },
      "#eMMC.SecureErase": {
        "@Redfish.ActionInfo": "/redfish/v1/Managers/BMC_0/Oem/EmmcSecureEraseActionInfo",
        "target": "/redfish/v1/Managers/BMC_0/Actions/Oem/eMMC.SecureErase"
      }
    }
  },
  "DateTime": "2025-04-17T07:32:29+00:00",
  "DateTimeLocalOffset": "+00:00",
  "Description": "Baseboard Management Controller",
  "EthernetInterfaces": {
    "@odata.id": "/redfish/v1/Managers/BMC_0/EthernetInterfaces"
  },
  "FirmwareVersion": "88.0002.1132",
  "Id": "BMC_0",
  "LastResetTime": "2025-05-19T11:29:55+00:00",
  "Links": {
    "ActiveSoftwareImage": {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/MGX_FW_BMC_0"
    },
    "ManagerForChassis": [
      {
        "@odata.id": "/redfish/v1/Chassis/BMC_eeprom"
      }
    ],
    "ManagerForChassis@odata.count": 1,
    "ManagerForServers": [
      {
        "@odata.id": "/redfish/v1/Systems/System_0"
      }
    ],
    "ManagerForServers@odata.count": 1,
    "ManagerInChassis": {
      "@odata.id": "/redfish/v1/Chassis/BMC_eeprom"
    },
    "SoftwareImages": [
      {
        "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/MGX_FW_BMC_0"
      }
    ],
    "SoftwareImages@odata.count": 1
  },
  "LogServices": {
    "@odata.id": "/redfish/v1/Managers/BMC_0/LogServices"
  },
  "ManagerDiagnosticData": {
    "@odata.id": "/redfish/v1/Managers/BMC_0/ManagerDiagnosticData"
  },
  "ManagerType": "BMC",
  "Model": "OpenBmc",
  "Name": "OpenBmc Manager",
  "NetworkProtocol": {
    "@odata.id": "/redfish/v1/Managers/BMC_0/NetworkProtocol"
  },
  "Oem": {
    "Nvidia": {
      "@odata.type": "#NvidiaManager.v1_6_0.NvidiaManager",
      "DebugTokenManagement": {
        "@odata.id": "/redfish/v1/Managers/BMC_0/Oem/Nvidia/DebugTokenManagement"
      },
      "FirmwareBuildType": null,
      "OTPProvisioned": false,
      "PersistentStorageSettings": {
        "Status": {
          "State": "Enabled"
        }
      }
    },
    "UptimeSeconds": 1419.15
  },
  "OpenBmc": {
```

	<pre> "@odata.id": "/redfish/v1/Managers/BMC_0#/Oem/OpenBmc", "@odata.type": "#OpenBMCManager.OpenBmc", "Certificates": { "@odata.id": "/redfish/v1/Managers/BMC_0/Truststore/Certificates" } }, "PowerState": "On", "SerialConsole": { "ConnectTypesSupported": ["IPMI", "SSH"], "MaxConcurrentSessions": 15, "ServiceEnabled": true }, "ServiceEntryPointUUID": "6df9822a-8b3d-4c85-b169-10870f0dc2d0", "ServiceIdentification": "", "Status": { "Conditions": [], "Health": "OK", "HealthRollup": "OK", "State": "Enabled" }, "UUID": "32b49574-4028-4c58-a70f-f1b9bf1ecaff" </pre>
Related Commands	Set Time and Date Manually, Enable/Disable NTP Servers
Notes	

3.8.4 Set Time and Date Manually

Redfish API	curl -k -u <user>:<p/w> https://<bmc_ip>/redfish/v1/Managers/BMC_0 -H content-type:application/json -X PATCH -d '{"DateTime": <date_time>}'	
Description	<p>This Redfish API shall be used to set the date and time on the BMC</p> <div style="border: 1px solid orange; padding: 5px; margin: 5px 0;">  Make sure to disable the NTP Servers before setting the date and time. </div>	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	date_time	Date/Time: i.e "2025-04-17T07:32:00.000Z"
Default	N/A	
History	88.0002.1040	
Response Example		
Related Commands	Show NTP Servers Status, Set Time and Date Manually	
Notes		

3.9 Network Interfaces

The Network Interfaces section is composed of the following APIs which work in conjunction:

- [Show Network Interface List](#)
- [Show Network Interface Details](#)
- [Set BMC Hostname](#)

3.9.1 Show Network Interface List

Redfish API	curl -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/Managers/BMC_0/EthernetInterfaces	
Description	This Redfish API shall be used to query for available network interfaces.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	
Response Example	<pre> { "@odata.id": "/redfish/v1/Managers/BMC_0/EthernetInterfaces", "@odata.type": "#EthernetInterfaceCollection.EthernetInterfaceCollection", "Description": "Collection of EthernetInterfaces for this Manager", "Members": [{ "@odata.id": "/redfish/v1/Managers/BMC_0/EthernetInterfaces/eth0" }, { "@odata.id": "/redfish/v1/Managers/BMC_0/EthernetInterfaces/eth1" }, { "@odata.id": "/redfish/v1/Managers/BMC_0/EthernetInterfaces/usb0" }], "Members@odata.count": 3, "Name": "Ethernet Network Interface Collection" } </pre>	
Related Commands	Show Network Interface Details	
Notes		

3.9.2 Show Network Interface Details

Redfish API	curl -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/Managers/BMC_0/EthernetInterfaces/<interface name>	
Description	This Redfish API shall be used query information about a specific BMC network interface.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	interface name	BMC network interface name: usb0, eth0
Default	N/A	
History	88.0002.0574	

Response Example	<pre> { "@odata.id": "/redfish/v1/Managers/BMC_0/EthernetInterfaces/usb0", "@odata.type": "#EthernetInterface.v1_9_0.EthernetInterface", "DHCIPv4": { "DHCPEnabled": false, "UseDNSServers": true, "UseDomainName": true, "UseNTPServers": true }, "DHCIPv6": { "OperatingMode": "Disabled", "UseDNSServers": true, "UseDomainName": true, "UseNTPServers": true }, "Description": "Management Network Interface", "EthernetInterfaceType": "Physical", "FQDN": "juliet-bmc", "HostName": "juliet-bmc", "IPv4Addresses": [{ "Address": "10.0.1.1", "AddressOrigin": "Static", "Gateway": "0.0.0.0", "SubnetMask": "255.255.255.0" }], "IPv4StaticAddresses": [{ "Address": "10.0.1.1", "AddressOrigin": "Static", "Gateway": "0.0.0.0", "SubnetMask": "255.255.255.0" }], "IPv6AddressPolicyTable": [], "IPv6Addresses": [{ "Address": "fe80::10c9:79ff:fe17:1c26", "AddressOrigin": "LinkLocal", "AddressState": "Preferred", "PrefixLength": 64 }], "IPv6DefaultGateway": "0:0:0:0:0:0:0:0", "IPv6StaticAddresses": [], "IPv6StaticDefaultGateways": [], "Id": "usb0", "InterfaceEnabled": true, "LinkStatus": "LinkUp", "MACAddress": "12:c9:79:17:1c:26", "MTUSize": 1500, "Name": "Manager Ethernet Interface", "NameServers": [], "SpeedMbps": 0, "StatelessAddressAutoConfig": { "IPv6AutoConfigEnabled": true }, "StaticNameServers": [], "Status": { "State": "Enabled" } } </pre>
Related Commands	Show Network Interface List
Notes	

3.9.3 Set BMC Hostname

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" https://<bmc_ip>/redfish/v1/Managers/BMC_0/EthernetInterfaces/<interface_name> -X PATCH -d '{"HostName": "<hostname>"}'	
Description	This Redfish API shall be used to query for available network interfaces.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	interface_name	BMC network interface name: usb0, eth0

	hostname	The desired Hostname
Default	N/A	
History	88.0002.1040	
Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }] } </pre>	
Related Commands	Show Network Interface Details	
Notes		

3.10 EEPROM

3.10.1 Show EEPROM Information

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/json" -X GET https://<bmc_ip>/redfish/v1/Chassis/BMC_eeprom	
Description	This Redfish API shall be used to provide information from BMC EEPROM	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	

<p>Response Example</p>	<pre> { "@odata.id": "/redfish/v1/Chassis/BMC_eeeprom", "@odata.type": "#Chassis.v1_22_0.Chassis", "Assembly": { "@odata.id": "/redfish/v1/Chassis/BMC_eeeprom/Assembly" }, "ChassisType": "Component", "Controls": { "@odata.id": "/redfish/v1/Chassis/BMC_eeeprom/Controls" }, "EnvironmentMetrics": { "@odata.id": "/redfish/v1/Chassis/BMC_eeeprom/EnvironmentMetrics" }, "Id": "BMC_eeeprom", "Links": { "ComputerSystems": [{ "@odata.id": "/redfish/v1/Systems/System_0" }], "ManagedBy": [{ "@odata.id": "/redfish/v1/Managers/BMC_0" }] }, "Location": { "PartLocation": { "LocationType": "Embedded" } }, "LogServices": { "@odata.id": "/redfish/v1/Chassis/BMC_eeeprom/LogServices" }, "Manufacturer": "NVIDIA", "Model": "P3809", "Name": "BMC_eeeprom", "Oem": { "Nvidia": { "@odata.type": "#NvidiaChassis.v1_1_0.NvidiaChassis" } }, "PCIeDevices": { "@odata.id": "/redfish/v1/Chassis/BMC_eeeprom/PCIeDevices" }, "PCIeSlots": { "@odata.id": "/redfish/v1/Chassis/BMC_eeeprom/PCIeSlots" }, "PartNumber": "699-13809-1404-500", "PowerState": "On", "PowerSubsystem": { "@odata.id": "/redfish/v1/Chassis/BMC_eeeprom/PowerSubsystem" }, "Sensors": { "@odata.id": "/redfish/v1/Chassis/BMC_eeeprom/Sensors" }, "SerialNumber": "1581324710134", "Status": { "Conditions": [], "Health": "OK", "HealthRollup": "OK", "State": "Enabled" }, "ThermalSubsystem": { "@odata.id": "/redfish/v1/Chassis/BMC_eeeprom/ThermalSubsystem" }, "TrustedComponents": { "@odata.id": "/redfish/v1/Chassis/BMC_eeeprom/TrustedComponents" } } </pre>
<p>Related Commands</p>	
<p>Notes</p>	

3.11 Temperature Sensor

3.11.1 Show Temperature Sensor Information

<p>Redfish API</p>	<p>curl -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/Chassis/MGX_BMC_0/Sensors/BMC_TEMP</p>
<p>Description</p>	<p>This Redfish API shall be used to read the value and thresholds from the BMC temperature sensor.</p>

Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	
Response Example	<pre> { "@odata.id": "/redfish/v1/Chassis/MGX_BMC_0/Sensors/BMC_TEMP", "@odata.type": "#Sensor.v1_2_0.Sensor", "Id": "BMC_TEMP", "Name": "BMC_TEMP", "Reading": 40.625, "ReadingRangeMax": 127.0, "ReadingRangeMin": -128.0, "ReadingType": "Temperature", "ReadingUnits": "Cel", "RelatedItem": [{ "@odata.id": "/redfish/v1/Systems/System_0" }], "Status": { "Conditions": [], "Health": "OK", "HealthRollup": "OK", "State": "Enabled" }, "Thresholds": { "LowerCaution": { "Reading": 5.0 }, "UpperCaution": { "Reading": 105.0 }, "UpperCritical": { "Reading": 108.0 } } } </pre>	
Related Commands		
Notes		

3.12 Service Identification

The Service Identification section is composed of the following APIs:

- [Set Service Identification](#)
- [Show Service Identification](#)

3.12.1 Set Service Identification

Redfish API	curl -k -u <user>:<p/w> -H 'Content-Type: application/json' -X PATCH https://<bmc_ip>/redfish/v1/Managers/BMC_0 -d '{"ServiceIdentification": "<service_id>"}	
Description	This Redfish API shall be used to set the ServiceIdentification property, which is a user-provided product and service identifier.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	service_id	The provided user ServiceIdentification (i.e., nvswitch_bmc_1)
Default	N/A	

History	88.0002.0929
Response Example	N/A
Related Commands	Show Service Identification
Notes	

3.12.2 Show Service Identification

Redfish API	curl -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/Managers/BMC_0	
Description	This Redfish API shall be used to read the ServiceIdentification property.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0929	

Response Example

```
{
  "@odata.id": "/redfish/v1/Managers/BMC_0",
  "@odata.type": "#Manager.v1_14_0.Manager",
  "Actions": {
    "#Manager.Reset": {
      "@Redfish.ActionInfo": "/redfish/v1/Managers/BMC_0/ResetActionInfo",
      "target": "/redfish/v1/Managers/BMC_0/Actions/Manager.Reset"
    },
    "#Manager.ResetToDefaults": {
      "ResetType@Redfish.AllowableValues": [
        "ResetAll"
      ],
      "target": "/redfish/v1/Managers/BMC_0/Actions/Manager.ResetToDefaults"
    },
    "Oem": {
      "#NvidiaManager.AsyncOOBRawCommand": {
        "@Redfish.ActionInfo": "/redfish/v1/Managers/BMC_0/Oem/Nvidia/
AsyncOOBRawCommandActionInfo",
        "target": "/redfish/v1/Managers/BMC_0/Actions/Oem/
NvidiaManager.AsyncOOBRawCommand"
      },
      "#NvidiaManager.ResetToDefaults": {
        "target": "/redfish/v1/Managers/BMC_0/Actions/Oem/
NvidiaManager.ResetToDefaults"
      },
      "#NvidiaManager.SyncOOBRawCommand": {
        "@Redfish.ActionInfo": "/redfish/v1/Managers/BMC_0/Oem/Nvidia/
SyncOOBRawCommandActionInfo",
        "target": "/redfish/v1/Managers/BMC_0/Actions/Oem/
NvidiaManager.SyncOOBRawCommand"
      },
      "#eMMC.SecureErase": {
        "@Redfish.ActionInfo": "/redfish/v1/Managers/BMC_0/Oem/
EmmcSecureEraseActionInfo",
        "target": "/redfish/v1/Managers/BMC_0/Actions/Oem/eMMC.SecureErase"
      }
    }
  },
  "DateTime": "2025-01-30T07:39:59+00:00",
  "DateTimeLocalOffset": "+00:00",
  "Description": "Baseboard Management Controller",
  "EthernetInterfaces": {
    "@odata.id": "/redfish/v1/Managers/BMC_0/EthernetInterfaces"
  },
  "FirmwareVersion": "88.0002.0109",
  "Id": "BMC_0",
  "LastResetTime": "2025-01-30T06:58:39+00:00",
  "Links": {
    "ActiveSoftwareImage": {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/MGX_FW_BMC_0"
    },
    "ManagerForChassis": [
      {
        "@odata.id": "/redfish/v1/Chassis/BMC_eeprom"
      }
    ],
    "ManagerForChassis@odata.count": 1,
    "ManagerForServers": [
      {
        "@odata.id": "/redfish/v1/Systems/System_0"
      }
    ],
    "ManagerForServers@odata.count": 1,
    "ManagerInChassis": {
      "@odata.id": "/redfish/v1/Chassis/BMC_eeprom"
    },
    "SoftwareImages": [
      {
        "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/MGX_FW_BMC_0"
      }
    ],
    "SoftwareImages@odata.count": 1
  },
  "LogServices": {
    "@odata.id": "/redfish/v1/Managers/BMC_0/LogServices"
  },
  "ManagerDiagnosticData": {
    "@odata.id": "/redfish/v1/Managers/BMC_0/ManagerDiagnosticData"
  },
  "ManagerType": "BMC",
  "Model": "OpenBmc",
  "Name": "OpenBmc Manager",
  "NetworkProtocol": {
    "@odata.id": "/redfish/v1/Managers/BMC_0/NetworkProtocol"
  },
  "Oem": {
    "Nvidia": {
      "@odata.type": "#NvidiaManager.v1_4_0.NvidiaManager",
      "FirmwareBuildType": null,
      "OTPProvisioned": false,
      "PersistentStorageSettings": {
        "Status": {
          "State": "Enabled"
        }
      }
    },
    "UptimeSeconds": 2411.55
  },
  "OpenBmc": {
    "@odata.id": "/redfish/v1/Managers/BMC_0/Oem/OpenBmc",
    "@odata.type": "#OpenBMCManager.OpenBmc",
  }
}
```

	<pre> "Certificates": { "@odata.id": "/redfish/v1/Managers/BMC_0/Truststore/Certificates" } }, "PowerState": "On", "SerialConsole": { "ConnectTypesSupported": ["IPMI", "SSH"], "MaxConcurrentSessions": 15, "ServiceEnabled": true }, "ServiceEntryPointUUID": "23d523e1-666f-4af5-b841-3241b2639516", "ServiceIdentification": "nvswitch_bmc_1", "Status": { "Conditions": [], "Health": "OK", "HealthRollup": "OK", "State": "Enabled" }, "UUID": "3330f3dd-cc81-4af3-8076-32f838318d3a" } </pre>
Related Commands	Set Service Identification
Notes	

3.13 Debug Information

The Debug Information section is composed of several APIs, some of which work in conjunction:

- [Generate Debug Information](#)
- [Show Debug Information Generation Status](#)
- [Output Debug Information To File](#)

These three APIs are related to generating debug information in the system.

The [Generate Debug Information](#) API initiate the task of generating debug information in the system. Following that task initiation one can call the [Show Debug Information Generation Status](#) API to figure out that task state (i.e., whether a task is still running or completed).

Upon completion of that task, one may choose to call the [Output Debug Information To File](#) API to see that relevant information outputted into a file.

3.13.1 Generate Debug Information

Redfish API	curl -k -u <user>:<p/w> -d '{"DiagnosticDataType": "Manager"}' -X POST https://<bmc_ip>/redfish/v1/Managers/BMC_0/LogServices/Dump/Actions/LogService.CollectDiagnosticData	
Description	This Redfish API shall be used to send a request for BMC to generate BMC dumps, ERoTs dumps, FPGAs dumps and BMC EEPROM dump.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	Id (return value)	task_id_number
Default	N/A	
History	88.0002.0574	

Response Example	<pre>{ "@odata.id": "/redfish/v1/TaskService/Tasks/0", "@odata.type": "#Task.v1_4_3.Task", "Id": "0", "TaskState": "Running", "TaskStatus": "OK" }</pre>
Related Commands	Show Debug Information Generation Status Output Debug Information To File
Notes	

3.13.2 Show Debug Information Generation Status

Redfish API	curl -k -u <user>:<p/w> -H 'Content-Type: application/json' -X GET https://<bmc_ip>/redfish/v1/TaskService/Tasks/<task_id_number>	
Description	This Redfish API shall be used to get the logs dump task state. When task state changes from "Running" to "Completed", the dump is ready for download.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	task_id_number	Task ID is received from Generate Debug Information command.
	entry id (return value)	entry_id
Default	N/A	
History	88.0002.0574	

Response Example	<pre> { "@odata.id": "/redfish/v1/TaskService/Tasks/0", "@odata.type": "#Task.v1_4_3.Task", "EndTime": "2024-03-25T11:56:11+00:00", "HidePayload": false, "Id": "0", "Messages": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The task with Id '0' has started.", "MessageArgs": ["0"], "MessageId": "TaskEvent.1.0.3.TaskStarted", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.16.0.Success", "MessageSeverity": "OK", "Resolution": "None" }], "Name": "Task 0", "Payload": { "HttpHeaders": ["Host: 10.0.1.1", "User-Agent: curl/7.88.1", "Accept: /*/*", "Content-Length: 33", "Location: /redfish/v1/Managers/BMC_0/LogServices/Dump/Entries/113"], "HttpOperation": "POST", "JsonBody": "{\n \"DiagnosticDataType\": \"Manager\"\n}", "TargetUri": "/redfish/v1/Managers/BMC_0/LogServices/Dump/Actions/LogService.CollectDiagnosticData" }, "PercentComplete": 100, "StartTime": "2024-03-25T11:44:30+00:00", "TaskMonitor": "/redfish/v1/TaskService/Tasks/0/Monitor", "TaskState": "Completed", "TaskStatus": "OK" } </pre>
Related Commands	Generate Debug Information Output Debug Information To File
Notes	

3.13.3 Output Debug Information To File

Redfish API	<pre>curl -k -u <user>:<p/w> -d '{"DiagnosticDataType": "Manager"}' -X POST https://<bmc_ip>/redfish/v1/Managers/BMC_0/LogServices/Dump/Entries/<entry_id>/attachment --output </path/to/tar/log_dump.tar.xz></pre>	
Description	This Redfish API shall be used to send a request for BMC to generate BMC dump as a compressed tar file, for a specific log entry ID. Use the entry ID from the previous RF API response.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	entry_id	Entry ID of the dump in redfish/v1/Managers/BMC_0/LogServices/Dump/Entries/ Use the entry id from the previous API. Show Debug Information Generation Status
	/path/to/tar/log_dump.tar.xz	path to download the log dump log_dump.tar.xz
Default	N/A	
History	88.0002.0574	

Response Example	<pre> % Total % Received % Xferd Average Speed Time Time Time Current Dload Upload Total Spent Left Speed 100 26933 100 26933 0 0 122k 0 --:--:-- --:--:-- --:--:-- 123k </pre>
Related Commands	<p>Generate Debug Information</p> <p>Show Debug Information Generation Status</p>
Notes	

3.14 Debug Token (CRDT)

The Debug Token section is composed of the following APIs, some of which work in conjunction:

- [Generate Debug Token](#)
- [Show Debug Token Generation Status](#)
- [Download Debug Token](#)
- [Install Debug Token Signed Firmware](#)
- [Show Debug Token Installation Status](#)
- [Generate Installed Debug Token Attachments](#)
- [Show Installed Debug Token Attachments Generation Status](#)
- [Show Installed Debug Token Attachments](#)

The first three APIs are related to generating debug token information in the system.

The [Generate Debug Token](#) API initiate the task of generating debug token information in the system. Following that task initiation, one can call the [Show Debug Token Generation Status](#) API to figure out that task state (i.e. whether still running or completed).

Upon completion of that task, one may choose to call the [Download Debug Token](#) API to see that relevant EROT SPDM information.

Once download the debug token, it can be used outside of the BMC platform to bundle it with a certain firmware package and to generate bundled debug file out of it.

The bundled and signed debug file shall be installed on the system using the API [Install Debug Token Signed Firmware](#), and another API [Show Debug Token Installation Status](#) can be used to verify the installation status.

The final three APIs are related to generating the installed debug token bundle information in the system.

The [Generate Installed Debug Token Attachments](#) API initiate a task of generating the installed debug token attachments in the system. Following that task initiation one can call the [Show Installed Debug Token Attachments Generation Status](#) API to figure out that task state (i.e., whether still running or completed).

Upon completion of that task, one may choose to call the [Show Installed Debug Token Attachments](#) API to see that relevant debug token attachment information.

3.14.1 Generate Debug Token

Redfish API	<pre> curl -k -u <user>:<p/w> --request POST --location 'https://<bmc_ip>/redfish/v1/Systems/System_0/LogServices/DebugTokenService/Actions/LogService.CollectDiagnosticData' \ --header 'Content-Type: application/json' \ --data '{ "DiagnosticDataType": "OEM", "OEMDiagnosticDataType": "GetDebugTokenRequest" }' </pre>
-------------	--

Description	This Redfish API shall be used to provide consolidated support for obtaining and querying Debug Tokens across the Juliet platform.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	Id (return value)	task_id_number
Default	N/A	
History	88.0002.0574	
Response Example	<pre>{ "@odata.id": "/redfish/v1/TaskService/Tasks/26", "@odata.type": "#Task.v1_4_3.Task", "Id": "26", "TaskState": "Running", "TaskStatus": "OK" }</pre>	
Related Commands	Show Debug Token Generation Status Download Debug Token Install Debug Token Signed Firmware Show Debug Token Installation Status Generate Installed Debug Token Attachments Show Installed Debug Token Attachments Generation Status Show Installed Debug Token Attachments	
Notes	<ul style="list-style-type: none"> • An EROT firmware greater than 183 is required. • The command provide a consolidated Install status and ability to collect Token Requests. • The system MUST have valid certificates on the EROT for testing 	

3.14.2 Show Debug Token Generation Status

Redfish API	curl -k -u <user>:<p/w> https://<bmc_ip>/redfish/v1/TaskService/Tasks/<task_id_number>	
Description	This Redfish API shall be used to show status of the Debug Tokens generation task state. When task state changes from “Running” to “Completed”, the Debug Token is ready to be downloaded.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	task_id_number	Task ID number provided from Generate Debug Token command
	attach_id (return value)	Entry ID number provided for the attachment
Default	N/A	
History	88.0002.0574	

Response Example

```
{
  "@odata.id": "/redfish/v1/TaskService/Tasks/26",
  "@odata.type": "#Task.v1_4_3.Task",
  "EndTime": "2024-07-04T05:10:50+00:00",
  "HidePayload": false,
  "Id": "26",
  "Messages": [
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The task with Id '26' has started.",
      "MessageArgs": [
        "26"
      ],
      "MessageId": "TaskEvent.1.0.3.TaskStarted",
      "MessageSeverity": "OK",
      "Resolution": "None."
    },
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The operation to request a debug token for device 'MGX_ERoT_BMC_0' has been successfully completed.",
      "MessageArgs": [
        "MGX_ERoT_BMC_0"
      ],
      "MessageId": "OpenBMC.0.4.0.DebugTokenRequestSuccess",
      "MessageSeverity": "OK",
      "Resolution": "None."
    },
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The operation to request a debug token for device 'MGX_ERoT_FPGA_0' has been successfully completed.",
      "MessageArgs": [
        "MGX_ERoT_FPGA_0"
      ],
      "MessageId": "OpenBMC.0.4.0.DebugTokenRequestSuccess",
      "MessageSeverity": "OK",
      "Resolution": "None."
    },
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The operation to request a debug token for device 'MGX_ERoT_NVSwitch_0' has been successfully completed.",
      "MessageArgs": [
        "MGX_ERoT_NVSwitch_0"
      ],
      "MessageId": "OpenBMC.0.4.0.DebugTokenRequestSuccess",
      "MessageSeverity": "OK",
      "Resolution": "None."
    },
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The operation to request a debug token for device 'MGX_ERoT_NVSwitch_1' has been successfully completed.",
      "MessageArgs": [
        "MGX_ERoT_NVSwitch_1"
      ],
      "MessageId": "OpenBMC.0.4.0.DebugTokenRequestSuccess",
      "MessageSeverity": "OK",
      "Resolution": "None."
    },
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The operation to request a debug token for device 'MGX_ERoT_CPU_0' has been successfully completed.",
      "MessageArgs": [
        "MGX_ERoT_CPU_0"
      ],
      "MessageId": "OpenBMC.0.4.0.DebugTokenRequestSuccess",
      "MessageSeverity": "OK",
      "Resolution": "None."
    },
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The task with Id '26' has completed.",
      "MessageArgs": [
        "26"
      ],
      "MessageId": "TaskEvent.1.0.3.TaskCompletedOK",
      "MessageSeverity": "OK",
      "Resolution": "None."
    }
  ],
  "Name": "Task 26",
  "Payload": {
    "HttpHeaders": [
      "Host: 10.0.1.1",
      "User-Agent: curl/7.88.1",
      "Accept: */*",
      "Content-Length: 78",
      "Location: /redfish/v1/Systems/System_0/LogServices/DebugTokenService/Entries/1/attachment"
    ],
    "HttpOperation": "POST",
    "JsonBody": "{\n  \"DiagnosticDataType\": \"OEM\",\n  \"OEMDiagnosticDataType\": \"GetDebugTokenRequest\"\n}",
    "TargetUri": "/redfish/v1/Systems/System_0/LogServices/DebugTokenService/LogService.CollectDiagnosticData"
  },
  "PercentComplete": 100,
}
```

	<pre>"StartTime": "2024-07-04T05:10:50+00:00", "TaskMonitor": "/redfish/v1/TaskService/Tasks/26/Monitor", "TaskState": "Completed", "TaskStatus": "OK" }</pre>
Related Commands	<p>Generate Debug Tokens Download Debug Token Install Debug Token Signed Firmware Show Debug Token Installation Status Generate Installed Debug Token Attachments Show Installed Debug Token Attachments Generation Status Show Installed Debug Token Attachments</p>
Notes	

3.14.3 Download Debug Token

Redfish API	curl -k -u <user>:<p/w> https://<bmc_ip>/redfish/v1/Systems/System_0/LogServices/DebugTokenService/Entries/<attach_id>/attachment -o </path/to/file.bin>	
Description	This Redfish API is used to download debug token request file generated.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	attach_id	ID for the attachment from Show Debug Token Generation Status command
	/path/to/file.bin	Path to download the debug token as bin file
Default	N/A	
History	88.0002.0574	
Response Example	<pre>curl -k -u root:0penBmc https://10.0.1.1/redfish/v1/Systems/System_0/LogServices/DebugTokenService/Entries/1/attachment -o /tmp/debug_token_request.bin % Total % Received % Xferd Average Speed Time Time Time Current Dload Upload Total Spent Left Speed 100 26933 100 26933 0 0 122k 0 ---:--:-- --:--:-- --:--:-- 123k</pre>	
Related Commands	<p>Generate Debug Tokens Show Debug Token Generation Status Install Debug Token Signed Firmware Show Debug Token Installation Status Generate Installed Debug Token Attachments Show Installed Debug Token Attachments Generation Status Show Installed Debug Token Attachments</p>	
Notes	After downloading the debug token, there needs to be a manual process of bundling the debug token and the firmware and signing them up via 3s servers.	

3.14.4 Install Debug Token Signed Firmware

Redfish API	curl -k -u <user>:<p/w> -H "Content-Type: application/octet-stream" -X POST https://<bmc_ip>/redfish/v1/UpdateService -T /tmp/<fw_pkg>	
Description	This Redfish API is used to install the signed debug token and firmware bundle.	
Syntax Description	user	BMC Username

	p/w	BMC User password
	bmc_ip	BMC IP address
	fw_pkg	Signed debug token and firmware bundle
	Id (return value)	task_id_number
Default	N/A	
History	88.0002.0574	
Response Example	<pre>{ "@odata.id": "/redfish/v1/TaskService/Tasks/2", "@odata.type": "#Task.v1_4_3.Task", "Id": "0", "TaskState": "Running", "TaskStatus": "OK" }</pre>	
Related Commands	Generate Debug Tokens Show Debug Token Generation Status Download Debug Token Show Debug Token Installation Status Generate Installed Debug Token Attachments Show Installed Debug Token Attachments Generation Status Show Installed Debug Token Attachments	
Notes	 Complete this step only after bundling the debug token request and debug firmware and getting it signed.	

3.14.5 Show Debug Token Installation Status

Redfish API	curl -k -u <user>:<p/w> https://<bmc_ip>/redfish/v1/TaskService/Tasks/<task_id_number>	
Description	This Redfish API shall be used to show status of the Debug Tokens Installation task state. When task state changes from “Running” to “Completed”, the Debug Token and Firmware bundled are installed.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	task_id_number	Task ID number provided from Install Debug Token Signed Firmware command
Default	N/A	
History	88.0002.0574	
Response Example	<pre>{ "@odata.id": "/redfish/v1/TaskService/Tasks/2", "@odata.type": "#Task.v1_4_3.Task", "Id": "0", "TaskState": "Running", "TaskStatus": "OK" }</pre>	

Related Commands	Generate Debug Tokens Show Debug Token Generation Status Download Debug Token Install Debug Token Signed Firmware Generate Installed Debug Token Attachments Show Installed Debug Token Attachments Generation Status Show Installed Debug Token Attachments
Notes	

3.14.6 Generate Installed Debug Token Attachments

Redfish API	<pre>curl -k -u <user>:<p/w> --request POST --location 'https://<bmc_ip>/redfish/v1/Systems/System_0/LogServices/DebugTokenService/Actions/LogService.CollectDiagnosticData' \ --header 'Content-Type: application/json' \ --data '{ "DiagnosticDataType": "OEM", "OEMDiagnosticDataType": "GetDebugTokenStatus" }'</pre>	
Description	This Redfish API is used to verify the installed debug bundle running state	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	Id (return value)	task_id_number
Default	N/A	
History	88.0002.0574	
Response Example	<pre>{ "@odata.id": "/redfish/v1/TaskService/Tasks/0", "@odata.type": "#Task.v1_4_3.Task", "Id": "0", "TaskState": "Running", "TaskStatus": "OK" }</pre>	
Related Commands	Generate Debug Tokens Show Debug Token Generation Status Download Debug Token Install Debug Token Signed Firmware Show Debug Token Installation Status Show Installed Debug Token Attachments Generation Status Show Installed Debug Token Attachments	
Notes		

3.14.7 Show Installed Debug Token Attachments Generation Status

Redfish API	<pre>curl -k -u <user>:<p/w> https://<bmc_ip>/redfish/v1/TaskService/Tasks/<task_id_number></pre>
Description	This Redfish API shall be used to show status of the Installed Debug Tokens Attachment generation task state. When task state changes from “Running” to “Completed”, the Installed Debug Token attachments are ready.

Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	task_id_number	Task ID number provided from Generate Installed Debug Token Attachments command
	attach_id	Entry ID for the attachment of EROT details and the programming state of the debug token bundle
Default	N/A	
History	88.0002.0574	

Response Example

See the Debug Token status

```
curl -k -u root:OpenBmc https://10.0.1.1/redfish/v1/TaskService/Tasks/0
{
  "@odata.id": "/redfish/v1/TaskService/Tasks/0",
  "@odata.type": "#Task.v1_4_3.Task",
  "EndTime": "2024-07-15T05:58:37+00:00",
  "HidePayload": false,
  "Id": "0",
  "Messages": [
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The task with Id '0' has started.",
      "MessageArgs": [
        "0"
      ],
      "MessageId": "TaskEvent.1.0.3.TaskStarted",
      "MessageSeverity": "OK",
      "Resolution": "None."
    },
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The operation to obtain a token status for device 'MGX_ERoT_BMC_0' has been successfully completed.",
      "MessageArgs": [
        "MGX_ERoT_BMC_0"
      ],
      "MessageId": "OpenBMC.0.4.0.DebugTokenStatusSuccess",
      "MessageSeverity": "OK",
      "Resolution": "None."
    },
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The operation to obtain a token status for device 'MGX_ERoT_FPGA_0' has been successfully completed.",
      "MessageArgs": [
        "MGX_ERoT_FPGA_0"
      ],
      "MessageId": "OpenBMC.0.4.0.DebugTokenStatusSuccess",
      "MessageSeverity": "OK",
      "Resolution": "None."
    },
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The operation to obtain a token status for device 'MGX_ERoT_NVSwitch_0' has been successfully completed.",
      "MessageArgs": [
        "MGX_ERoT_NVSwitch_0"
      ],
      "MessageId": "OpenBMC.0.4.0.DebugTokenStatusSuccess",
      "MessageSeverity": "OK",
      "Resolution": "None."
    },
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The operation to obtain a token status for device 'MGX_ERoT_NVSwitch_1' has been successfully completed.",
      "MessageArgs": [
        "MGX_ERoT_NVSwitch_1"
      ],
      "MessageId": "OpenBMC.0.4.0.DebugTokenStatusSuccess",
      "MessageSeverity": "OK",
      "Resolution": "None."
    },
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The operation to obtain a token status for device 'MGX_ERoT_CPU_0' has been successfully completed.",
      "MessageArgs": [
        "MGX_ERoT_CPU_0"
      ],
      "MessageId": "OpenBMC.0.4.0.DebugTokenStatusSuccess",
      "MessageSeverity": "OK",
      "Resolution": "None."
    },
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The task with Id '0' has completed.",
      "MessageArgs": [
        "0"
      ],
      "MessageId": "TaskEvent.1.0.3.TaskCompletedOK",
      "MessageSeverity": "OK",
      "Resolution": "None."
    }
  ],
  "Name": "Task 0",
  "Payload": {
    "HttpHeaders": [
      "Host: 10.0.1.1",
      "User-Agent: curl/7.88.1",
      "Accept: */*",
      "Content-Length: 74",
      "Location: /redfish/v1/Systems/System_0/LogServices/DebugTokenService/Entries/0/attachment"
    ],
    "HttpOperation": "POST",
    "JsonBody": "{\n  \"DiagnosticDataType\": \"OEM\",\n  \"OEMDiagnosticDataType\": \"DebugTokenStatus\"\n}"
  }
}
```

	<pre> "TargetUri": "/redfish/v1/Systems/System_0/LogServices/DebugTokenService/ LogService.CollectDiagnosticData" }, "PercentComplete": 100, "StartTime": "2024-07-15T05:58:36+00:00", "TaskMonitor": "/redfish/v1/TaskService/Tasks/0/Monitor", "TaskState": "Completed", "TaskStatus": "OK" } </pre>
Related Commands	Generate Debug Tokens Show Debug Token Generation Status Download Debug Token Install Debug Token Signed Firmware Show Debug Token Installation Status Generate Installed Debug Token Attachments Show Installed Debug Token Attachments
Notes	

3.14.8 Show Installed Debug Token Attachments

Redfish API	curl -k -u <user>:<p/w> https://<bmc_ip>/redfish/v1/Systems/System_0/LogServices/DebugTokenService/Entries/<attach_id>/attachment	
Description	This Redfish API is used to show the Installed Debug Token Attachments.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	attach_id	ID for the attachment from Show Installed Debug Token Attachments Generation Status command
Default	N/A	
History	88.0002.0574	
Response Example	See the Debug Token status <pre> curl -k -u root:OpenBmc https://10.0.1.1/redfish/v1/Systems/System_0/LogServices/ DebugTokenService/Entries/0/attachment 00 00 16 47 00 01 0F 01 00 00 02 1E 03 1A 18 0F 07 0E 01 00 00 16 47 00 01 0F 01 00 00 05 1E 03 1A 18 10 29 2D 01 00 00 16 47 00 01 0F 01 00 00 01 1E 03 1A 18 10 25 0A 01 00 00 16 47 00 01 0F 01 00 00 05 1E 03 1A 18 10 25 18 01 00 00 16 47 00 01 0F 01 00 00 05 1E 03 1A 18 10 30 26 01 10'th Byte: Token installation status(1 for installed and 0 for not installed) 11-18 Byte: ERoT Serial Number 19th Byte: Tells it is Prod system(2) or debug system(1). </pre>	
Related Commands	Generate Debug Tokens Show Debug Token Generation Status Download Debug Token Install Debug Token Signed Firmware Show Debug Token Installation Status Generate Installed Debug Token Attachments Show Installed Debug Token Attachments Generation Status	
Notes		

3.15 Leakage Sensor

3.15.1 Register to Leakage Events

You can subscribe for Redfish leakage events from anywhere in the network BMC is connected to.

Instructions for event subscription are as follows (bear in mind this is just an example, there are many other methods to do that):

1. git clone <https://github.com/DMTF/Redfish-Event-Listener.git>
2. cd Redfish-Event-Listener
3. sudo pip install -r requirements.txt
4. Update the config.ini file with your listener IP and BMC IP.
5. sudo python RedfishEventListener_v1.py
6. Simulate Leakage event on BMC.
7. You should start receiving leakage events in console.

Example for config.ini:

```
[Information]
Updated = February 24, 2023
Description = Redfish Event Listener Tool Simple Config

[SystemInformation]
ListenerIP = 10.210.25.179
ListenerPort = 12345
UseSSL = off

[CertificateDetails]
certfile = cert.pem
keyfile = server.key

[SubscriptionDetails]
Destination = https://<ListenerIP>/
EventTypes = [
    "Alert",
    "ResourceRemoved",
    "ResourceAdded",
    "ResourceUpdated",
    "StatusChange"]
Context = Public
Format = Event
Expand = false
ResourceTypes = ["Chassis"]
Registries = ["ResourceEvent"]

[ServerInformation]
ServerIPs = ["https://10.209.51.198:443"]
UserNames = ["root"]
Passwords = ["0penBmc"]
LoginType = ["Session"]
```

Example of what leakage events look like:

```
{
  "@odata.type": "#Event.v1_9_0.Event",
  "Context": "",
  "Events": [
    {
      "EventTimeStamp": "2024-06-18T15:17:39+00:00",
      "LogEntry": {
        "@odata.id": "/redfish/v1/Systems/System_0/LogServices/EventLog/Entries/24297"
      },
      "Message": "The resource property leakage1 has detected errors of type 'Leakage Detected'.",
      "MessageArgs": [
        "leakage1",
        "Leakage Detected"
      ],
      "MessageId": "ResourceEvent.1.0.ResourceErrorsDetected",
      "MessageSeverity": "Critical",
      "Oem": {
        "Nvidia": {
          "@odata.type": "#NvidiaEvent.v1_0_0.EventRecord",
          "Device": "",
          "ErrorId": ""
        }
      }
    }
  ]
}
```

```

    },
    "Resolution": "Inspect for water leakage and consider power down switch tray."
  }
},
"Id": "1",
"Name": "Event Log"}

```

3.15.2 Leakage Sensor Status

Redfish API	curl -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/Chassis/MGX_BMC_0/ThermalSubsystem/LeakDetection/LeakDetectors/<arg>	
Description	This Redfish API is used to show the status of the leakage sensors.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	arg	leakage1 to leakage6
Default	N/A	
History	88.0002.0754	
Response Example	<pre> { "@odata.id": "/redfish/v1/Chassis/MGX_BMC_0/ThermalSubsystem/LeakDetection/LeakDetectors/leakage6", "@odata.type": "#LeakDetector.v1_0_1.LeakDetector", "DetectorState": "OK", "Id": "leakage6", "LeakDetectorType": "Moisture", "Name": "Leak Detector", "Status": { "Health": "OK", "State": "Enabled" } } </pre>	
Related Commands		
Notes		

3.16 Factory Reset

BMC factory reset can be done in two manners:

- [Factory Reset \(Configuration Only\)](#)
- [Factory Reset \(Configuration & Logs\)](#)

3.16.1 Factory Reset (Configuration Only)

Redfish API	curl -k -u <user>:<p/w> -X POST https://<bmc_ip>/redfish/v1/Managers/BMC_0/Actions/Manager.ResetToDefaults -d '{"ResetToDefaultsType": "ResetAll"}'	
Description	This Redfish API shall be used to "Factory Reset" the BMC Configuration (Users, P/W, Network, and so forth) to the initial state in which it came from production. Following the Redfish command, BMC will reboot and apply the factory reset. Upon login, the user will be prompted to change the default factory password.	
Syntax Description	user	BMC Username
	p/w	BMC User password

	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	
Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.16.0.Success", "MessageSeverity": "OK", "Resolution": "None" }] } </pre>	
Related Commands	Factory Reset (Configuration & Logs)	
Notes		

3.16.2 Factory Reset (Configuration & Logs)

Redfish API	curl -k -u <user>:<p/w> -H 'Content-Type:application/json' -X POST -d '{"ResetToDefaultsType": "ResetAll"}' https://<bmc_ip>/redfish/v1/Managers/BMC_0/Actions/Oem/NvidiaManager.ResetToDefaults	
Description	<p>This Redfish API shall be used to "Factory Reset" the BMC configuration (Users, P/W, Network, and so forth) to the initial state in which it came from production. It will also delete all logs created by BMC.</p> <p>Following this Redfish command, BMC will reboot and apply the factory reset. Upon login, the user will be prompted to change the default factory password.</p>	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0574	
Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.16.0.Success", "MessageSeverity": "OK", "Resolution": "None" }] } </pre>	
Related Commands	Factory Reset (Configuration Only)	
Notes		

3.17 eMMC Secure Erase

3.17.1 Securely Erase eMMC Card

Redfish API	curl -k -u <user>:<p/w> -X POST https://<bmc_ip>/redfish/v1/Managers/BMC_0/Actions/Oem/eMMC.SecureErase	
Description	This Redfish API shall be used to securely wipe all data stored on the eMMC card of the BMC. Following this Redfish command, BMC will reboot and apply the eMMC secure erase.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.1040	
Response Example	<pre>{ "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }] }</pre>	
Related Commands	Factory Reset Commands	
Notes		

3.18 Rsyslog

The BMC can stream out local logs (that go to the systemd journal) by using rsyslog.



- Make sure to configure <rsyslog_server_ip> and <port> according to the server-side configuration.
- This manual will not cover the configuration of the rsyslog server.
- The BMC here acts as an rsyslog client.

- [Configure Rsyslog Client](#)
- [Query Rsyslog Client Configuration](#)
- [Enable Encrypted Streaming With TLS](#)
- [Change Rsyslog Transport Protocol](#)
- [Configure Facility Filters](#)
- [Configure Priority Filters](#)

3.18.1 Configure Rsyslog Client

Redfish API	curl -k -u <user>:<p/w> -X PATCH https://<bmc_ip>/redfish/v1/Managers/BMC_0/NetworkProtocol -d '{ "Oem": { "Nvidia": { "Rsyslog": { "Address": "<rsyslog_server_ip>", "Port": <rsyslog_server_port>, "State": "Enabled" } } } }'	
Description	This Redfish API shall be used to configure the rsyslog client on BMC side.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	rsyslog_server_ip	The rsyslog server IP
	rsyslog_server_port	The rsyslog server Port
Default	N/A	
History	88.0002.0929	
Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }] } </pre>	
Related Commands	Query Rsyslog Client Configuration RSYSLOG settings commands	
Notes		

3.18.2 Query Rsyslog Client Configuration

Redfish API	curl -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/Managers/BMC_0/NetworkProtocol	
Description	This Redfish API shall be used to query the rsyslog client configuration on BMC side.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0929	
Response Example	N/A	
Related Commands	Configure Rsyslog Client Configure Rsyslog	

Notes	
-------	--

3.18.3 Enable Encrypted Streaming With TLS

Redfish API	curl -k -u <user>:<p/w> -X PATCH https://<bmc_ip>/redfish/v1/Managers/BMC_0/NetworkProtocol -d '{"Oem": {"Nvidia": {"Rsyslog": {"TLS": "Enabled"} } } }'	
Description	This Redfish API shall be used to enable TLS encrypted streaming for rsyslog messages.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
Default	N/A	
History	88.0002.0929	
Response Example	<pre>{ "@odata.id": "/redfish/v1/Managers/BMC_0/NetworkProtocol", "@odata.type": "#ManagerNetworkProtocol.v1_5_0.ManagerNetworkProtocol", "Description": "Manager Network Service", "FQDN": "juliet-bmc", "HTTP": { "Port": null, "ProtocolEnabled": false }, "HTTPS": { "Certificates": { "@odata.id": "/redfish/v1/Managers/BMC_0/NetworkProtocol/HTTPS/Certificates" }, "Port": 443, "ProtocolEnabled": true }, "HostName": "juliet-bmc", "Id": "NetworkProtocol", "Name": "Manager Network Protocol", "Oem": { "Nvidia": { "@odata.type": "#NvidiaNetworkProtocol.v1_0_0.NetworkProtocol", "Rsyslog": { "Address": "10.210.25.179", "Filter": { "Facilities": ["All"], "LowestSeverity": "All" }, "Port": 6514, "State": "Enabled", "TLS": "Enabled", "TransportProtocol": "TCP" } } }, "SSH": { "Port": 22, "ProtocolEnabled": true }, "Status": { "Health": "OK", "HealthRollup": "OK", "State": "Enabled" } }</pre>	
Related Commands	Query Rsyslog Client Configuration	
Notes		

3.18.4 Change Rsyslog Transport Protocol

Redfish API	curl -k -u <user>:<p/w> -X PATCH https://<bmc_ip>/redfish/v1/Managers/BMC_0/NetworkProtocol -d '{ "Oem": { "Nvidia": { "Rsyslog": { "TransportProtocol": "<protocol>", "Address": "10.0.1.2", "Port": 512, "State": "Enabled" } } } }'	
Description	This Redfish API shall be used in order to select UDP or TCP rsyslog transport protocol.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	protocol	TCP or UDP
Default	TCP	
History	88.0002.1301	
Response Example	<pre> { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }, { "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }] </pre>	
Related Commands	Use the following command to retrieve current Transport Protocol: curl -k -u <user>:<p/w> -X GET https://<bmc_ip>/redfish/v1/Managers/BMC_0/NetworkProtocol	
Notes		

3.18.5 Configure Facility Filters

Redfish API	curl -k -u <user>:<p/w> -X PATCH https://<bmc_ip>/redfish/v1/Managers/BMC_0/NetworkProtocol -d '{ "Oem": { "Nvidia": { "Rsyslog": { "Filter": { "Facilities": ["<facility>"] } } } } }'
Description	This Redfish API shall be used to configure the rsyslog facilities filter used during message streaming. The facility filter value can be Daemon for system service, kern for kernel and ALL for all messages.

Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	facility	Possible values: All, Kern, Daemon
Default	N/A	
History	88.0002.0929	
Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }] } </pre>	
Related Commands	Query Rsyslog Client Configuration	
Notes		

3.18.6 Configure Priority Filters

Redfish API	curl -k -u <user>:<p/w> -X PATCH https://<bmc_ip>/redfish/v1/Managers/BMC_0/NetworkProtocol -d '{ "Oem": { "Nvidia": { "Rsyslog": { "Filter": { "LowestSeverity": "<severity>" } } } } }	
Description	This Redfish API shall be used to enable streaming of log messages that their severity is equal/higher than the configured LowestSeverity.	
Syntax Description	user	BMC Username
	p/w	BMC User password
	bmc_ip	BMC IP address
	severity	Possible values: Error, Warning, Info, All
Default	N/A	
History	88.0002.0929	
Response Example	<pre> { "@Message.ExtendedInfo": [{ "@odata.type": "#Message.v1_1_1.Message", "Message": "The request completed successfully.", "MessageArgs": [], "MessageId": "Base.1.18.1.Success", "MessageSeverity": "OK", "Resolution": "None." }] } </pre>	

Related Commands	Query Rsyslog Client Configuration
Notes	

3.19 Supported Redfish Scheme URIs

Schema Reference	Redfish URIs
AccountService	/redfish/v1/AccountService/Accounts
	/redfish/v1/AccountService/Roles
	/redfish/v1/AccountService/Roles/Administrator
	/redfish/v1/AccountService/Roles/Operator
	/redfish/v1/AccountService/Roles/ReadOnly
CertificateService	/redfish/v1/CertificateService
	/redfish/v1/CertificateService/CertificateLocations
Chassis	/redfish/v1/Chassis
	/redfish/v1/Chassis/BMC_eeeprom
	/redfish/v1/Chassis/CPLD_XX
	/redfish/v1/Chassis/MGX_XX
	/redfish/v1/Chassis/MGX_ERoT_XX
	/redfish/v1/Chassis/MGX_ERoT_XXX/Certificates
	/redfish/v1/Chassis/MGX_ERoT_XXX/Certificates/CertChain
	/redfish/v1/Chassis/MGX_BMC_0/Sensors/BMC_TEMP
	/redfish/v1/Chassis/MGX_BMC_0/ThermalSubsystem/LeakDetection
	/redfish/v1/Chassis/MGX_BMC_0/ThermalSubsystem/LeakDetection/LeakDetectors/leakageX
	ComponentIntegrity
/redfish/v1/ComponentIntegrity/MGX_ERoT_XX	
/redfish/v1/ComponentIntegrity/MGX_ERoT_XX/Actions/ComponentIntegrity.SPDMGetSignedMeasurements	
EventService	/redfish/v1/EventService
Managers	/redfish/v1/Managers
	/redfish/v1/Managers/BMC_0
	/redfish/v1/Managers/BMC_0/Actions

Schema Reference	Redfish URIs
	/redfish/v1/Managers/BMC_0/Actions/Manager.ResetToDefaults
	/redfish/v1/Managers/BMC_0/EthernetInterfaces
	/redfish/v1/Managers/BMC_0/EthernetInterfaces/XXX
	/redfish/v1/Managers/BMC_0/ManagerDiagnosticData
	/redfish/v1/Managers/BMC_0/NetworkProtocol
	/redfish/v1/Managers/BMC_0/LogServices/Dump/Actions
	/redfish/v1/Managers/BMC_0/NetworkProtocol/HTTPS/Certificates
SessionService	/redfish/v1/SessionService
Systems	/redfish/v1/Systems
	/redfish/v1/Systems/System_0
	/redfish/v1/Systems/System_0/Actions/ComputerSystem.Reset
	/redfish/v1/Systems/System_0/LogServices
	/redfish/v1/Systems/System_0/LogServices/FaultLog
	/redfish/v1/Systems/System_0/LogServices/EventLog
	/redfish/v1/Systems/System_0/LogServices/Dump
	/redfish/v1/Systems/System_0/LogServices/DebugTokenService
TaskService	/redfish/v1/TaskService
	/redfish/v1/TaskService/Tasks
UpdateService	/redfish/v1/UpdateService
	/redfish/v1/UpdateService/FirmwareInventory

✘ Unsupported URIs

Please note, the following URIs are functional but not supported by the switch BMC firmware:

Schema Reference	Redfish URIs
Registries	/redfish/v1/Registries
	/redfish/v1/Registries/Base
	/redfish/v1/Registries/TaskEvent
	/redfish/v1/Registries/ResourceEvent

Schema Reference	Redfish URIs
	/redfish/v1/Registries/OpenBMC
	/redfish/v1/Registries/Telemetry
	/redfish/v1/Registries/Platform
	/redfish/v1/Registries/Update
JsonSchemas	/redfish/v1/JsonSchemas

4 Document Revision History

88.0002.1301—July 2025

Added:

- [Change Rsyslog Transport Protocol](#)

88.0002.1139—June 2025

Added:

- [Show NTP Servers Status](#)
- [Enable/Disable NTP Servers](#)
- [Show Time and Date](#)
- [Set Time and Date Manually](#)

88.0002.1040—May 2025

Added:

- [Updating Firmware with Multipart API](#)
- [Show Update Firmware Multipart Status](#)
- [CPU Host Reset Action Information](#)
- [Apply BMC Manager Reset](#)
- [BMC Host Reset Action Information](#)
- [Set BMC Hostname](#)
- [Securely Erase eMMC Card](#)
- [Show Minimum Security Version Information for Application Firmware](#)

Updated

- Reset types in [Apply CPU Host Reset](#)

88.0002.0956—April 2025

Added:

- The URI [/redfish/v1/Chassis/MGX_BMC_0/ThermalSubsystem/LeakDetection](#)
- [Create New BMC User](#)
- [Change BMC User Account Permissions](#)
- [Change BMC Account Lockout Duration](#)
- [Change BMC Minimum Password Length](#)
- [Change BMC Account Lockout Threshold](#)

88.0002.0931—February 2025

Added:

- The subsection [Service Identification](#)
- The subsection [Rsyslog](#)
- The section [Supported Redfish Scheme URIs](#)

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. Neither NVIDIA Corporation nor any of its direct or indirect subsidiaries and affiliates (collectively: "NVIDIA") make any representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice. Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA, the NVIDIA logo, and Mellanox are trademarks and/or registered trademarks of NVIDIA Corporation and/or its



affiliates in the U.S. and in other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2025 NVIDIA Corporation & affiliates. All Rights Reserved.

