



Monitoring

Table of contents

System FRU	3
System Logs	6
BMC Sensor Data	83
BlueField Arm State	93
Rsyslog	95
DPU Chassis	103
DPU Information	109
BMC and BlueField Logs	110
System Processor	121

This section contains the following pages:

- [System FRU](#)
- [System Logs](#)
- [BMC Sensor Data](#)
- [BlueField Arm State](#)
- [Rsyslog](#)
- [DPU Chassis](#)
- [DPU Information](#)
- [BMC and BlueField Logs](#)
- [System Processor](#)

System FRU

FRU Reading Redfish Commands

FRU data is embedded within the chassis schema. To retrieve the relevant FRU data, execute the following Redfish command:

```
curl -k -u root:'<password>' -H 'Content-Type: application/json'
-X GET https://<bmc_ip>/redfish/v1/Chassis/Card1/
```

The FRU data can be found in the following read attributes:

```
...
  "Manufacturer": "https://www.mellanox.com",
  "Model": "BlueField-3 DPU",
...
  "PartNumber": "900-9D3D4-00EN-HAR",
...
  "SerialNumber": "MT2421XZ0H DU",
...
```

FRU Reading IPMI Commands

To retrieve FRU info, run:

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN fru print
```

System FRU ID 0 contains information for the NVIDIA® BlueField® device:

i Note

The following is supported when operating in DPU mode only.

```
FRU Device Description : Builtin FRU Device (ID 0)
Chassis Type           : Main Server Chassis
Chassis Part Number    : 900-9D3D4-00EN-HAA
Chassis Serial         : N/A
Chassis Extra          : N/A
Chassis Extra          : ..
Chassis Extra          : https://www.mellanox.com
Board Mfg Date         : Mon Aug  5 16:39:00 2024
Board Mfg              : https://www.mellanox.com
Board Product          : BlueField-3 DPU
Board Serial           : N/A
Board Part Number      : 900-9D3D4-00EN-HAA
Board Extra            : ..
Product Manufacturer   : https://www.mellanox.com
Product Name           : BlueField-3 DPU
Product Part Number    : 900-9D3D4-00EN-HAA
Product Version        : N/A
Product Serial         : MT2430XZ0A14
Product Asset Tag      : N/A
Product Extra          : ..
```

```
FRU Device Description : Nvidia-BMCMezz (ID 169)
Board Mfg Date         : Mon Aug  5 16:39:00 2024
Board Mfg              : Nvidia
Board Product          : Nvidia-BMCMezz
Board Serial           : MT2430XZ0A14
Board Part Number      : 900-9D3D4-00EN-HAA
```

To print a specific FRU:

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN fru print  
<fru_id>
```

To dump the binary FRU data into a file:

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN fru read  
<fru_id> <filename>
```

Note

The parameter `<filename>` is the absolute path to the file.

Info

Using the `ipmitool fru` command displays all the FRU devices detected by the BMC.

System Logs

System Event Logs

The System Event Log (SEL) and Event Log in OpenBMC provide robust mechanisms for monitoring, diagnosing, and troubleshooting hardware and system issues.

- SEL
 - Functionality – The SEL captures and records significant system events related to hardware and firmware. This includes events such as hardware failures, temperature thresholds, power anomalies, and other critical system changes.
 - Access – The SEL can be accessed via IPMI\Redfish commands, allowing administrators to query and retrieve logs for analysis
 - Management – Administrators can clear, save, and manage SEL entries to maintain system health and ensure critical events are recorded accurately
- Event Log:
 - Functionality – The Event Log provides a comprehensive record of both hardware and software events, offering detailed insights into system operations and potential issues. This includes firmware updates, configuration changes, security alerts, etc.
 - Access – The Event Log is accessible via Redfish interface, enabling easy retrieval and management of event data
 - Management – Users can filter, sort, and analyze events to identify patterns, diagnose problems, and improve system reliability. The Event Log supports exporting logs for offline analysis and archiving.
- Key features
 - Scalability – Both the SEL and Event Log are designed to handle a high volume of events, ensuring no critical information is lost
 - Integration – These logs integrate seamlessly with existing management tools, providing a unified view of system health and events

- Usability – User-friendly interfaces and command-line tools make it easy to access and manage logs, ensuring administrators can quickly respond to issues

Overall, the SEL and Event Log in OpenBMC are essential tools for maintaining system integrity, improving reliability, and ensuring swift resolution of any issues that arise.

Event Log Redfish Commands

Displaying Event Log Information

```
curl -k -u root:'<password>' -H 'Content-Type: application/json'
-X GET
https://<bmc_ip>/redfish/v1/Systems/Bluefield/LogServices/EventLog
```

Output example:

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog",
  "@odata.type": "#LogService.v1_1_0.LogService",
  "Actions": {
    "#LogService.ClearLog": {
      "target":
      "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Actions/LogServiceClearLog"
    }
  },
  "DateTime": "2023-09-27T14:28:50+00:00",
  "DateTimeLocalOffset": "+00:00",
  "Description": "System Event Log Service",
  "Entries": {
    "@odata.id":
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries"
  },
  "Id": "EventLog",
  "Name": "Event Log Service",
```

```

"Oem": {
  "Nvidia": {
    "@odata.type": "#NvidiaLogService.v1_0_0.NvidiaLogService",
    "LatestEntryID": "4",
    "LatestEntryTimeStamp": "2023-09-27T14:19:30+00:00"
  }
},
"OverWritePolicy": "WrapsWhenFull"
}

```

Displaying List of Events

```

curl -k -u root:'<password>' -H 'Content-Type: application/json'
-X GET
https://<bmc_ip>/redfish/v1/Systems/Bluefield/LogServices/EventLog,

```

Output example:

```

{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries",
  "@odata.type": "#LogEntryCollection.LogEntryCollection",
  "Description": "Collection of System Event Log Entries",
  "Members": [
    {
      "@odata.id":
      "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/1",
      "@odata.type": "#LogEntry.v1_9_0.LogEntry",
      "AdditionalDataURI":
      "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/1/attach",
      "Created": "2023-09-27T14:18:39+00:00",
      "EntryType": "Event",

```

```

    "Id": "1",
    "Message": "12V_ATX sensor crossed a warning low threshold
going low. Reading=6.048000 Threshold=10.400000.",
    "MessageArgs": [
        "12V_ATX",
        "6.048000",
        "10.400000"
    ],
    "MessageId":
"OpenBMC.0.1.SensorThresholdWarningLowGoingLow",
    "Name": "System Event Log Entry",
    "Resolution": "",
    "Resolved": false,
    "Severity": "OK"
}
...
],
"Members@odata.count": 1,
"Name": "System Event Log Entries"
}

```

Clearing Event Log

```

curl -k -u root:'<password>' -H 'Content-Type: application/json'
-X POST
https://<bmc_ip>/redfish/v1/Systems/Bluefield/LogServices/EventLog,

```

SEL Redfish Commands

Displaying SEL Information

```
curl -k -u root:'<password>' -H 'Content-Type: application/json'
-X GET
https://<bmc_ip>/redfish/v1/Systems/Bluefield/LogServices/SEL/
```

Output example:

```
{
  "@odata.id": "/redfish/v1/Systems/Bluefield/LogServices/SEL",
  "@odata.type": "#LogService.v1_1_0.LogService",
  "Actions": {
    "#LogService.ClearLog": {
      "target":
"/redfish/v1/Systems/Bluefield/LogServices/SEL/Actions/LogService.(
    }
  },
  "DateTime": "2024-07-18T10:54:52+00:00",
  "DateTimeLocalOffset": "+00:00",
  "Description": "IPMI SEL Service",
  "Entries": {
    "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/SEL/Entries"
  },
  "Id": "SEL",
  "Name": "SEL Log Service",
  "OverWritePolicy": "WrapsWhenFull"
}
```

Displaying List of Events

```
curl -k -u root:'<password>' -H 'Content-Type: application/json'
-X GET
```

https://<bmc_ip>/redfish/v1/Systems/Bluefield/LogServices/SEL/Entr:

Output example:

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/SEL/Entries",
  "@odata.type": "#LogEntryCollection.LogEntryCollection",
  "Description": "Collection of System Event Log Entries",
  "Members": [
    {
      "@odata.id":
      "/redfish/v1/Systems/Bluefield/LogServices/SEL/Entries/1",
      "@odata.type": "#LogEntry.v1_13_0.LogEntry",
      "Created": "2024-07-16T15:34:32+00:00",
      "EntryType": "SEL",
      "Id": "1",
      "Message": "12V_ATX sensor crossed a warning low threshold
going low. Reading=6.048000 Threshold=10.400000.",
      "MessageArgs": [
        "12V_ATX",
        "6.048000",
        "10.400000"
      ],
      "MessageId":
      "OpenBMC.0.1.SensorThresholdWarningLowGoingLow",
      "Name": "System Event Log Entry",
      "Resolution": "Check the sensor or subsystem for errors.",
      "Resolved": false,
      "Severity": "OK"
    },
    ...
  ],
  "Members@odata.count": 22,
```

```
"Name": "System Event Log Entries"
}
```

Clearing Event Log

```
curl -k -u root:'<password>' -H 'Content-Type: application/json'
-X POST
https://<bmc_ip>/redfish/v1/Systems/Bluefield/LogServices/EventLog,
```

Configuring SEL Info Log Capacity

```
curl -k -u root:'<password>' -H 'Content-Type: application/json'
-X POST
https://<bmc_ip>/redfish/v1/Managers/Bluefield_BMC/Actions/Oem/Nvidia
-d '{"ErrorInfoCap":300 }'
```

Getting SEL Info Log Capacity

```
curl -k -u root:'<password>' -H 'Content-Type: application/json'
-X GET
https://<bmc_ip>/redfish/v1/Managers/Bluefield_BMC/Oem/Nvidia/SelCa
```

Example output:

```
{
  "ErrorInfoCap": 300
}
```

```
}
```

SEL IPMI Commands

The following table lists the command to use to view event logs:

Displaying SEL Information

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN sel
```

Displaying List of Events

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN sel list
```

Displaying Extended List of Events

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN sel elist
```

Saving SEL Events to File

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN sel save  
<filename>
```

Clearing SEL

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN sel clear
```

Configuring SEL Info Log Capacity

The capacity is a 4-byte value, and the byte order is from low to high as shown in command example.

To set the capacity to 300 lines, the value should be `0x2c 0x01 0x00 0x00`:

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN raw  
0x0a 0x4a <capacity[0:7]> <capacity[8:15]> <capacity[16:23]>  
<capacity[24:31]>
```

Getting SEL Info Log Capacity

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN raw 0x0a  
0x4b
```

SEL Message Types

The following subsections detail the messages which are added to the BMC SEL and the scenarios that trigger them.

UEFI Boot

Messages are added to the BMC SEL while the BlueField UEFI is booting which describe the status of the UEFI boot.

SEL messages:

- SMBus initialization
- PCI resource configuration
- System boot initiated

Example:

```
SEL Record ID      : 0037
Record Type        : 02
Timestamp          : 06:36:06 UTC 06:36:06 UTC
Generator ID       : 0001
EvM Revision       : 04
Sensor Type        : System Firmware
Sensor Number      : 06
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
Event Data         : c207ff
Description        : PCI resource configuration
```

DPU Watchdog Sensor

A watchdog message will be added in the event of a DPU reset caused by the hardware watchdog.

Example:

```
10 | 07/17/25 | 12:43:16 UTC | Watchdog2 #0x28 | Power cycle |
Asserted
```

IPMB Sensors

QSFP Sensors

Messages are added to the SEL in case of a change in the status of the QSFP cables. The messages describe the event and status of the sensor.

List of QSFP sensors:

- `P0_link` – the QSFP 0 cable status
- `P1_link` – the QSFP 1 cable status

SEL messages:

- `Config Error` – the QSFP cable is down
- `Connected` – the QSFP cable is up

Example:

```
SEL Record ID      : 003e
Record Type        : 02
Timestamp          : 07:08:28 UTC 07:08:28 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Cable / Interconnect
Sensor Number      : 00
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
Event Data (RAW)   : 010f0f
Event Interpretation : Missing
Description        : Config Error

Sensor ID          : p0_link (0x0)
Entity ID          : 31.1
Sensor Type (Discrete): Cable / Interconnect
States Asserted    : Cable / Interconnect
                   [Config Error]
```

Temperature Sensors

Messages are added to the SEL if temperature sensors detect a value higher than the sensor thresholds. The messages include a description of the event, BlueField FRU device description, BlueField BMC device description, and the status of the sensor.

List of temperature sensors:

- `bluefield_temp` – Bluefield temperature
- `p0_temp` – QSFP 0 cable temperature
- `p1_temp` – QSFP 1 cable temperature
- `ddr_temp` – DDR temperature

SEL messages:

- `Upper Critical going high` – crossing a upper critical threshold
- `Upper Non-critical going high` – crossing a upper non-critical threshold
- `Lower Critical going low` – crossing a lower critical threshold
- `Lower Non-critical going low` – crossing a lower non-critical threshold

Example:

```
SEL Record ID      : 003c
Record Type        : 02
Timestamp          : 07:01:06 UTC 07:01:06 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Temperature
Sensor Number      : 03
Event Type         : Threshold
Event Direction    : Assertion Event
Event Data (RAW)   : 592802
Trigger Reading    : 40.000degrees C
```

```

Trigger Threshold      : 2.000degrees C
Description            : Upper Critical going high

Sensor ID              : p0_temp (0x3)
Entity ID              : 0.1
Sensor Type (Threshold) : Temperature
Sensor Reading         : 40 (+/- 0) degrees C
Status                 : ok
Lower Non-Recoverable : na
Lower Critical         : -5.000
Lower Non-Critical    : 0.000
Upper Non-Critical    : 70.000
Upper Critical        : 75.000
Upper Non-Recoverable : na
Positive Hysteresis   : Unspecified
Negative Hysteresis   : Unspecified
Assertion Events      :
Event Enable          : Event Messages Disabled
Assertions Enabled    : lnc- lcr- unc+ ucr+
Deassertions Enabled : lnc+ lcr+ unc- ucr-

FRU Device Description : Nvidia-BMCMezz (ID 169)
Board Mfg Date         : Tue Jan  3 23:16:00 2023 UTC
Board Mfg              : Nvidia
Board Product          : Nvidia-BMCMezz
Board Serial           : MT2251XZ02W5
Board Part Number      : 900-9D3B6-00CV-AAA

FRU Device Description : BlueField-3 Smar (ID 250)
Board Mfg Date         : Tue Jan  3 23:16:00 2023 UTC
Board Mfg              : Nvidia
Board Product          : BlueField-3 SmartNIC Main Card
Board Serial           : MT2251XZ02W5
Board Part Number      : 900-9D3B6-00CV-AAA
Product Manufacturer   : Nvidia
Product Name           : BlueField-3 SmartNIC Main Card

```

```
Product Part Number : 900-9D3B6-00CV-AAA
Product Version     : A3
Product Serial      : MT2251XZ02W5
Product Asset Tag   : 900-9D3B6-00CV-AAA
```

Voltage Sensors

Messages are added to the SEL if a voltage sensor's reading crosses the sensor's thresholds. The messages include a description of the event, BlueField BMC device description, and the status of the sensor.

List of voltage sensors:

- `rtc_voltage` – RTC battery voltage

SEL messages:

- `Lower Critical going low` – crossing a lower critical threshold

Example:

```
SEL Record ID      : 0227
Record Type        : 02
Timestamp          : 02/17/25 16:01:21 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Voltage
Sensor Number      : 1a
Event Type         : Threshold
Event Direction    : Assertion Event
Event Data (RAW)   : 52004d
Trigger Reading    : 2.000Volts
Trigger Threshold  : 2.302Volts
Description        : Lower Critical going low
Sensor ID          : rtc_voltage (0x1a)
```

```
Entity ID           : 0.1
Sensor Type (Threshold) : Voltage
Sensor Reading      : 3.000 (+/- 0) Volts
Status              : ok
Lower Non-Recoverable : na
Lower Critical      : 2.302
Lower Non-Critical   : na
Upper Non-Critical   : na
Upper Critical       : na
Upper Non-Recoverable : na
Positive Hysteresis  : Unspecified
Negative Hysteresis  : Unspecified
Assertion Events     :
Event Enable         : Event Messages Disabled
Assertions Enabled    : lcr- ucr+
Deassertions Enabled : lcr+ ucr-

FRU Device Description : Nvidia-BMCMezz (ID 169)
Board Mfg Date         : Sun Feb 12 07:48:00 2023 UTC
Board Mfg              : Nvidia
Board Product          : Nvidia-BMCMezz
Board Serial           : MT2306XZ00BU
Board Part Number      : 900-9D3B6-00CC-AAA
Board Area Checksum    : OK
```

Power Sensors

Messages are added to the SEL if power sensors detect a value higher/lower than the sensor thresholds. The messages include a description of the event, BlueField BMC device description, and the status of the sensor.

List of power sensors:

- `soc_power` – current power consumption of the SoC
- `power_envelope` – maximum power consumption allowed to the SoC

SEL messages:

- Upper Non-critical going high – crossing an upper non-critical threshold (only in power_envelope)
- Lower Critical going low – crossing a lower critical threshold (only in soc_power)
- Lower Non-critical going low – crossing a lower non-critical threshold (only in power_envelope)

Example:

```
SEL Record ID      : 000e
Record Type        : 02
Timestamp          : 02/13/25 09:09:11 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Other
Sensor Number      : 05
Event Type         : Threshold
Event Direction    : Assertion Event
Event Data (RAW)   : 520005
Trigger Reading    : 0.000Watts
Trigger Threshold  : 5.000Watts
Description        : Lower Critical going low

Sensor ID          : soc_power (0x5)
Entity ID         : 0.1
Sensor Type (Threshold) : Other
Sensor Reading     : 0 (+/- 0) Watts
Status            : Lower Critical
Lower Non-Recoverable : na
Lower Critical     : 5.000
Lower Non-Critical : na
Upper Non-Critical : na
```

```
Upper Critical          : na
Upper Non-Recoverable  : na
Positive Hysteresis    : Unspecified
Negative Hysteresis    : Unspecified
Assertion Events       : lcr-
Event Enable           : Event Messages Disabled
Assertions Enabled     : lcr- ucr+
Deassertions Enabled   : lcr+ ucr-
```

```
FRU Device Description : Nvidia-BMCMezz (ID 169)
Board Mfg Date         : Mon Aug  7 07:48:00 2023 UTC
Board Mfg              : Nvidia
Board Product          : Nvidia-BMCMezz
Board Serial           : MT2329XZ010Z
Board Part Number      : 900-9D3B4-00EN-EAA
Board Area Checksum    : OK
```

Synthesized Sensors

Power Deviation Sensors

Messages are added to the SEL if power sensors detect a value higher or lower than the sensor thresholds.

List of power deviation sensors:

- `power_envelope_deviation` - Measure the deviation between the values of the `soc_power` and `power_envelope` sensors

SEL messages:

- `Upper Critical going high` - crossing a upper critical threshold
- `Upper Non-critical going high` - crossing a upper non-critical threshold

Example:

SEL Record ID : 0014
Record Type : 02
Timestamp : 02/13/25 09:17:11 UTC
Generator ID : 0020
EvM Revision : 04
Sensor Type : Other
Sensor Number : 04
Event Type : Threshold
Event Direction : Assertion Event
Event Data (RAW) : 590f05
Trigger Reading : 15.000Watts
Trigger Threshold : 5.000Watts
Description : Upper Critical going high

Sensor ID : power_envelope_d (0x4)
Entity ID : 0.1
Sensor Type (Threshold) : Other
Sensor Reading : 15 (+/- 0) Watts
Status : Upper Critical
Lower Non-Recoverable : na
Lower Critical : na
Lower Non-Critical : na
Upper Non-Critical : 0.000
Upper Critical : 5.000
Upper Non-Recoverable : na
Positive Hysteresis : Unspecified
Negative Hysteresis : Unspecified
Assertion Events : unc+ ucr+
Event Enable : Event Messages Disabled
Assertions Enabled : lnc- lcr- unc+ ucr+
Deassertions Enabled : lnc+ lcr+ unc- ucr-

FRU Device Description : Nvidia-BMCMezz (ID 169)
Board Mfg Date : Mon Aug 7 07:48:00 2023 UTC
Board Mfg : Nvidia

```
Board Product      : Nvidia-BMCMezz
Board Serial       : MT2329XZ010Z
Board Part Number  : 900-9D3B4-00EN-EAA
Board Area Checksum : OK
```

ADC Sensors

Messages are added to the SEL if the sensor voltage crosses the sensor's thresholds. The messages include a description of the event, BlueField FRU device description, BlueField BMC device description, and the status of the sensor.

List of ADC sensors:

- 1V_BMC
- 1_2V_BMC
- 1_8V
- 1_8V_BMC
- 2_5V
- 3_3V
- 3_3V_RGM
- 5V
- 12V_ATX
- 12V_PCIE
- DVDD
- HVDD

- VDD
- VDDQ
- VDD_CPU_L
- VDD_CPU_R

SEL messages:

- Upper Non-critical going high – crossing a upper non-critical threshold
- Lower Non-critical going low – crossing a lower non-critical threshold

Example:

```

SEL Record ID      : 0042
Record Type        : 02
Timestamp          : 09:20:50 UTC 09:20:50 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Voltage
Sensor Number      : 06
Event Type         : Threshold
Event Direction    : Assertion Event
Event Data (RAW)   : 50a9ff
Trigger Reading    : 1.200Volts
Trigger Threshold  : 1.810Volts
Description        : Lower Non-critical going low

Sensor ID          : 1_2V_BMC (0x6)
Entity ID          : 0.1
Sensor Type (Threshold) : Voltage
Sensor Reading     : 1.200 (+/- 0) Volts
Status            : ok
Lower Non-Recoverable : na
Lower Critical     : na

```

Lower Non-Critical : 1.143
Upper Non-Critical : 1.257
Upper Critical : na
Upper Non-Recoverable : na
Positive Hysteresis : Unspecified
Negative Hysteresis : Unspecified
Assertion Events :
Event Enable : Event Messages Disabled
Assertions Enabled : lnc- unc+
Deassertions Enabled : lnc+ unc-

FRU Device Description : Nvidia-BMCMezz (ID 169)
Board Mfg Date : Tue Jan 3 23:16:00 2023 UTC
Board Mfg : Nvidia
Board Product : Nvidia-BMCMezz
Board Serial : MT2251XZ02W5
Board Part Number : 900-9D3B6-00CV-AAA

FRU Device Description : BlueField-3 Smar (ID 250)
Board Mfg Date : Tue Jan 3 23:16:00 2023 UTC
Board Mfg : Nvidia
Board Product : BlueField-3 SmartNIC Main Card
Board Serial : MT2251XZ02W5
Board Part Number : 900-9D3B6-00CV-AAA
Product Manufacturer : Nvidia
Product Name : BlueField-3 SmartNIC Main Card
Product Part Number : 900-9D3B6-00CV-AAA
Product Version : A3
Product Serial : MT2251XZ02W5
Product Asset Tag : 900-9D3B6-00CV-AAA

System Commands

Warm Rebooting BlueField

SEL messages:

```
System boot initiated  
Initiated by warm reset
```

Example:

```
SEL Record ID      : 0001  
Record Type       : 02  
Timestamp         : 01/10/24 14:25:07 UTC  
Generator ID      : 0020  
EvM Revision      : 04  
Sensor Type       : System Boot Initiated  
Sensor Number     : 17  
Event Type        : Sensor-specific Discrete  
Event Direction   : Assertion Event  
Event Data        : 020000  
Description       : Initiated by warm reset
```

Hard Rebooting BlueField

SEL messages:

```
System boot initiated  
Initiated by hard reset
```

Example:

```
SEL Record ID      : 0008
Record Type        : 02
Timestamp          : 01/10/24 14:33:01 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : System Boot Initiated
Sensor Number      : 17
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
Event Data         : 010000
Description        : Initiated by hard reset
```

If the host does not assert the `PERST / ALL_STANDBY` signal, causing the reset to fail, the following SEL messages can be observed:

```
Power Unit
Failure detected
```

Example:

```
SEL Record ID      : 0004
Record Type        : 02
Timestamp          : 07/25/24 13:32:18 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Power Unit
Sensor Number      : 1b
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
Event Data         : 060000
```

Description : Failure detected

Shutting Down BlueField

SEL messages:

```
OS Critical Stop
OS graceful shutdown
```

Example:

```
SEL Record ID      : 000a
Record Type        : 02
Timestamp          : 01/10/24 14:34:45 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : OS Critical Stop
Sensor Number      : 18
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
Event Data         : 030000
Description        : OS graceful shutdown
```

Updating BlueField BFB Image

SEL messages:

```
Firmware or software change success
```

Example:

```
SEL Record ID      : 0007
Record Type        : 02
Timestamp          : 06/11/24 14:03:02 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Version Change
Sensor Number      : 18
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
Event Data         : c70000
Description        : Firmware or software change success
```

Updating BMC

SEL messages:

```
Firmware or software change success, Mngmt SW agent change
```

Example:

```
SEL Record ID      : 0010
Record Type        : 02
Timestamp          : 01/10/24 15:48:01 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Version Change
Sensor Number      : 19
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
```

```
Event Data           : c70e00
Description          : Firmware or software change success,
Mngmt SW agent change
```

RAS Errors

Multi-bit ECC

SEL messages:

```
Uncorrectable ECC
```

Example:

```
SEL Record ID       : 024a
Record Type         : 02
Timestamp           : 06/20/24 15:54:58 UTC
Generator ID        : 0020
EvM Revision        : 04
Sensor Type         : Memory
Sensor Number       : 17
Event Type          : Sensor-specific Discrete
Event Direction     : Assertion Event
Event Data          : 010000
Description         : Uncorrectable ECC
```

Single-bit ECC

SEL messages:

Correctable ECC

Example:

```
SEL Record ID      : 0254
Record Type        : 02
Timestamp          : 06/20/24 16:01:05 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Memory
Sensor Number      : 17
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
Event Data         : 000000
Description        : Correctable ECC
```

Cache Correctable Error

- Event `data1 0x0C` indicates Correctable machine check error
- Event `data2 0x1` indicates a cache error (same Processor Error Type enumeration used by Cper)

https://uefi.org/specs/UEFI/2.10_A/Apx_N_Common_Platform_Error_Record.html

SEL messages:

```
Correctable machine check error
```

Example:

```
SEL Record ID      : 009d
Record Type        : 02
Timestamp          : 12/17/24 12:16:35 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Processor
Sensor Number      : 18
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
Event Data         : 0c0100
Description        : Correctable machine check error
```

Cache Uncorrectable Error

- Event `data1 0x0C` indicates Correctable machine check error
- Event `data2 0x1` indicates a cache error (same Processor Error Type enumeration used by Cper)

https://uefi.org/specs/UEFI/2.10_A/Apx_N_Common_Platform_Error_Record.html

SEL messages:

```
Uncorrectable machine check exception
```

Example:

```
SEL Record ID      : 0012
Record Type        : 02
Timestamp          : 12/10/24 16:32:27 UTC
Generator ID       : 0020
EvM Revision       : 04
```

```
Sensor Type           : Processor
Sensor Number         : 1b
Event Type            : Sensor-specific Discrete
Event Direction       : Assertion Event
Event Data             : 0b0100
Description           : Uncorrectable machine check exception
```

Cache Uncorrectable Fatal Error

- Event `data1 0x0C` indicates Correctable machine check error
- Event `data2 0x1` indicates a cache error (same Processor Error Type enumeration used by Cper)
- Event `data3 0x1` indicates a fatal error .

https://uefi.org/specs/UEFI/2.10_A/Apx_N_Common_Platform_Error_Record.html

SEL messages:

```
Uncorrectable machine check exception
```

Example:

```
SEL Record ID        : 00b1
Record Type          : 02
Timestamp            : 12/17/24 16:07:11 UTC
Generator ID         : 0020
EvM Revision         : 04
Sensor Type          : Processor
Sensor Number        : 18
Event Type           : Sensor-specific Discrete
Event Direction      : Assertion Event
```

```
Event Data          : 0b0101
Description         : Uncorrectable machine check exception
```

PCIe Correctable Error

SEL messages:

```
Bus Correctable error
```

Example:

```
SEL Record ID      : 000c
Record Type        : 02
Timestamp          : 02/10/25 15:11:22 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Critical Interrupt
Sensor Number      : ff
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
Event Data         : 070000
Description        : Bus Correctable error
```

PCIe Uncorrectable Error

SEL messages:

```
Bus Uncorrectable error
```

Example:

```
SEL Record ID      : 001c
Record Type        : 02
Timestamp          : 02/12/25 09:30:22 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Critical Interrupt
Sensor Number      : ff
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
Event Data         : 080000
Description        : Bus Uncorrectable error
```

PCIe Fatal Error

SEL messages:

```
Bus Fatal Error
```

Example:

```
SEL Record ID      : 0012
Record Type        : 02
Timestamp          : 02/12/25 12:10:25 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Critical Interrupt
Sensor Number      : ff
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
```

```
Event Data          : 0a0000
Description         : Bus Fatal Error
```

ATX Power Error

SEL messages:

```
Power Supply
Failure detected
```

Example:

```
SEL Record ID      : 0006
Record Type        : 02
Timestamp          : 02/17/25 13:47:28 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Power Supply
Sensor Number      : 02
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
Event Data (RAW)   : 010000
Event Interpretation : Missing
Description        : Failure detected
```

Arm Frequency Change

The system's frequency is dynamically managed by the Arm cores, based on the system's power consumption and temperature. As long as they stay below a predefined threshold, the Arm cores operate at full frequency. If power consumption or temperature exceeds

their threshold, the frequency is reduced in stages for mitigation. This reduction will put the system under the crossed threshold, and then the frequency will be throttled back to full performance.

SEL message:

```
Throttled | Asserted
```

Example:

```
SEL Record ID      : 0004
Record Type        : 02
Timestamp          : 09/01/24 09:12:34 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Processor
Sensor Number      : ff
Event Type         : Sensor-specific Discrete
Event Direction    : Assertion Event
Event Data         : 0a0000
Description        : Throttled
```

Info

More details can be extracted from Redfish. Further information is available in section "[Redfish Event Log](#)".

Data Port Module Events

Data Port Module High Power Consumption Notification

An SEL entry is generated when the power consumption of a data port module exceeds a critical threshold.

SEL messages:

```
Voltage <sensor-id> | Upper Non-recoverable going high | Asserted
```

Example:

```
SEL Record ID      : 0029
Record Type        : 02
Timestamp          : 09/29/24 13:22:44 UTC
Generator ID       : 0020
EvM Revision       : 04
Sensor Type        : Voltage
Sensor Number      : 1d
Event Type         : Threshold
Event Direction    : Assertion Event
Event Data         : 0b0000
Description        : Upper Non-recoverable going high
```

Info

The sensor ID can be found using `ipmitool sdr list all -vv`.

- Port 0 sensor name: `voltage_p0`
- Port 1 sensor name: `voltage_p1`

Data Port Module Thermal "Going High" Notification

Indicates that the temperature of the data port module exceeded valid range.

SEL messages:

```
Temperature <sensor-id> | Upper Non-critical going high |  
Asserted
```

Example:

```
SEL Record ID      : 002c  
Record Type       : 02  
Timestamp         : 10/01/24 06:47:54 UTC  
Generator ID      : 0020  
EvM Revision      : 04  
Sensor Type       : Temperature  
Sensor Number     : 1d  
Event Type        : Threshold  
Event Direction   : Assertion Event  
Event Data        : 070000  
Description       : Upper Non-critical going high
```

Info

The sensor ID can be found using `ipmitool sdr list all -vv`.

- Port 0 sensor name: `thermal_p0`

- Port 1 sensor name: `thermal_p1`

Data Port Module Thermal "Going Low" Notification

Indicates that the temperature of the data port module returned to valid range.

SEL messages:

```
Temperature <sensor-id> | Upper Non-critical going low |  
Asserted
```

Example:

```
SEL Record ID      : 002d  
Record Type       : 02  
Timestamp         : 10/01/24 06:47:58 UTC  
Generator ID      : 0020  
EvM Revision      : 04  
Sensor Type       : Temperature  
Sensor Number     : 1d  
Event Type        : Threshold  
Event Direction   : Assertion Event  
Event Data        : 060000  
Description       : Upper Non-critical going low
```

Info

The sensor ID can be found using `ipmitool sdr list all -vv`.

- Port 0 sensor name: `thermal_p0`

- Port 1 sensor name: thermal_p1

Redfish Event Log

System Commands

Warm Rebooting BlueField

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/2",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/2/attach",
  "Created": "2024-07-26T10:41:57+00:00",
  "EntryType": "Event",
  "Id": "2",
  "Message": "DPU Warm Reset",
  "Modified": "2024-07-26T10:41:57+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Hard Rebooting BlueField

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/7",
```

```

    "@odata.type": "#LogEntry.v1_13_0.LogEntry",
    "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/7/attach
    "Created": "2024-07-26T09:50:16+00:00",
    "EntryType": "Event",
    "Id": "7",
    "Message": "DPU Hard Reset",
    "Modified": "2024-07-26T09:50:16+00:00",
    "Name": "System Event Log Entry",
    "Resolved": false,
    "Severity": "OK"
}

```

If the host does not assert the PERST signal, causing the reset to fail:

```

{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/8",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/8/attach
  "Created": "2024-07-26T09:58:34+00:00",
  "EntryType": "Event",
  "Id": "8",
  "Message": "PERST is in de-assert, skip SoC Hard Reset",
  "Modified": "2024-07-26T09:58:34+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}

```

If the host does not assert the `ALL_STANDBY` signal, causing the reset to fail:

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/8",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/8/attach",
  "Created": "2024-07-26T09:58:34+00:00",
  "EntryType": "Event",
  "Id": "8",
  "Message": "ALL_STDBY is in de-assert, skip SoC Hard
Reset",
  "Modified": "2024-07-26T09:58:34+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Shutting Down BlueField

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/18",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/18/attach",
  "Created": "2024-07-26T13:56:46+00:00",
  "EntryType": "Event",
  "Id": "18",
  "Message": "DPU Shutdown",
  "Modified": "2024-07-26T13:56:46+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
}
```

```
    "Severity": "OK"
  },
```

Updating BMC

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/2",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/2/attach",
  "Created": "2024-07-26T10:41:57+00:00",
  "EntryType": "Event",
  "Id": "2",
  "Message": "BMC SW update",
  "Modified": "2024-07-26T10:41:57+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
},
```

Getting Measurements

```
{
  "@odata.id": "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/12",
  "@odata.type": "#LogEntry.v1_15_0.LogEntry",
  "Created": "2025-03-04T15:34:43+00:00",
  "EntryType": "Event",
  "Id": "12",
  "Message": "Redfish attestation measurements POST request received",
  "Modified": "2025-03-04T15:34:43+00:00",
```

```
"Name": "System Event Log Entry",  
"Resolved": false,  
"Severity": "OK"  
}
```

Adding BMC User

```
{  
  "@odata.id":  
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/3",  
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",  
  "AdditionalDataURI":  
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/3/attach",  
  "Created": "2024-01-10T14:25:14+00:00",  
  "EntryType": "Event",  
  "Id": "3",  
  "Message": "BMC User Create test0",  
  "Modified": "2024-01-10T14:25:14+00:00",  
  "Name": "System Event Log Entry",  
  "Resolved": false,  
  "Severity": "OK"  
}
```

Deleting BMC User

```
{  
  "@odata.id":  
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/2",  
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",  
  "AdditionalDataURI":  
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/2/attach",  
}
```

```
"Created": "2024-01-10T14:25:14+00:00",
"EntryType": "Event",
"Id": "2",
"Message": "BMC User Delete test0",
"Modified": "2024-01-10T14:25:14+00:00",
"Name": "System Event Log Entry",
"Resolved": false,
"Severity": "OK"
}
```

Renaming BMC User

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/2",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/2/attach",
  "Created": "2024-01-10T14:25:14+00:00",
  "EntryType": "Event",
  "Id": "2",
  "Message": "BMC User Rename test0 To test1",
  "Modified": "2024-01-10T14:25:14+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

BMC User Login

```
{
```

```
    "@odata.id":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/27",  
    "@odata.type": "#LogEntry.v1_13_0.LogEntry",  
    "AdditionalDataURI":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/27/att  
    "Created": "2024-06-11T13:07:34+00:00",  
    "EntryType": "Event",  
    "Id": "27",  
    "Message": "User (root) logged in",  
    "Modified": "2024-06-11T13:07:34+00:00",  
    "Name": "System Event Log Entry",  
    "Resolved": false,  
    "Severity": "OK"  
  }
```

BMC User Logout

```
{  
  "@odata.id":  
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/37",  
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",  
  "AdditionalDataURI":  
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/37/att  
  "Created": "2024-06-11T13:30:48+00:00",  
  "EntryType": "Event",  
  "Id": "37",  
  "Message": "User (root) logged out",  
  "Modified": "2024-06-11T13:30:48+00:00",  
  "Name": "System Event Log Entry",  
  "Resolved": false,  
  "Severity": "OK"  
}
```

Changing BMC User Password

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/11",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/11/attach",
  "Created": "2024-06-11T13:03:42+00:00",
  "EntryType": "Event",
  "Id": "11",
  "Message": "Password changed for root",
  "Modified": "2024-06-11T13:03:42+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Changing BlueField UEFI Password

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/7",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/7/attach",
  "Created": "2024-06-11T13:02:04+00:00",
  "EntryType": "Event",
  "Id": "7",
  "Message": "Password changed for UEFI",
  "Modified": "2024-06-11T13:02:04+00:00",
}
```

```
"Name": "System Event Log Entry",
"Resolved": false,
"Severity": "OK"
}
```

Adding BMC IP Address

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/20",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/20/att
  "Created": "2024-07-25T13:40:22+00:00",
  "EntryType": "Event",
  "Id": "20",
  "Message": "BMC IP Address Added",
  "Modified": "2024-07-25T13:40:22+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
},
```

Deleting BMC IP Address

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/21",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/21/att
```

```
"Created": "2024-01-10T15:53:57+00:00",
"EntryType": "Event",
"Id": "21",
"Message": "BMC IP Address Deleted",
"Modified": "2024-01-10T15:53:57+00:00",
"Name": "System Event Log Entry",
"Resolved": false,
"Severity": "OK"
}
```

Changing BMC IPv4 Mode to Static

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/6",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/6/attach",
  "Created": "2024-06-11T13:02:04+00:00",
  "EntryType": "Event",
  "Id": "6",
  "Message": "Set IPv4 to Static mode",
  "Modified": "2024-06-11T13:02:04+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
},
```

Changing BMC IPv4 Mode to DHCP

```
{
```

```

    "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/9",
    "@odata.type": "#LogEntry.v1_13_0.LogEntry",
    "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/9/attach
    "Created": "2024-06-11T13:02:05+00:00",
    "EntryType": "Event",
    "Id": "9",
    "Message": "Set IPv4 to DHCP mode",
    "Modified": "2024-06-11T13:02:05+00:00",
    "Name": "System Event Log Entry",
    "Resolved": false,
    "Severity": "OK"
}

```

Changing BMC IPv6 Mode to Static

```

{
    "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/38",
    "@odata.type": "#LogEntry.v1_13_0.LogEntry",
    "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/38/attach
    "Created": "2024-06-11T13:34:57+00:00",
    "EntryType": "Event",
    "Id": "38",
    "Message": "Set IPv6 to Static mode",
    "Modified": "2024-06-11T13:34:57+00:00",
    "Name": "System Event Log Entry",
    "Resolved": false,
    "Severity": "OK"
}

```

Changing BMC IPv6 Mode to DHCP

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/39",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/39/attach",
  "Created": "2024-06-11T13:35:03+00:00",
  "EntryType": "Event",
  "Id": "39",
  "Message": "Set IPv6 to DHCP mode",
  "Modified": "2024-06-11T13:35:03+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Changing BMC NTP Server

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/8",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/8/attach",
  "Created": "2024-06-11T14:07:30+00:00",
  "EntryType": "Event",
  "Id": "8",
  "Message": "BMC NTP Servers Changed",
  "Modified": "2024-06-11T14:07:30+00:00",
}
```

```
"Name": "System Event Log Entry",
"Resolved": false,
"Severity": "OK"
}
```

Starting RShim on BMC

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/4",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/4/attach",
  "Created": "2024-06-11T13:00:41+00:00",
  "EntryType": "Event",
  "Id": "4",
  "Message": "Started rshim service on BMC",
  "Modified": "2024-06-11T13:00:41+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Stopping RShim on BMC

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/35",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/35/attach"
```

```
"Created": "2024-06-11T13:29:19+00:00",
"EntryType": "Event",
"Id": "35",
"Message": "Stopped rshim service on BMC",
"Modified": "2024-06-11T13:29:19+00:00",
"Name": "System Event Log Entry",
"Resolved": false,
"Severity": "OK"
}
```

Reset of TOR E-Switch

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/32",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/32/att
  "Created": "2024-06-11T13:19:57+00:00",
  "EntryType": "Event",
  "Id": "32",
  "Message": "Reset of TOR E-Switch",
  "Modified": "2024-06-11T13:19:57+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Setting Mode of 3-port Switch Ports to Allow All Ports to Access OOB RJ45

```
{
```

```
    "@odata.id":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/34",  
    "@odata.type": "#LogEntry.v1_13_0.LogEntry",  
    "AdditionalDataURI":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/34/att  
    "Created": "2024-06-11T13:20:12+00:00",  
    "EntryType": "Event",  
    "Id": "34",  
    "Message": "All ports are allowed access to RJ45",  
    "Modified": "2024-06-11T13:20:12+00:00",  
    "Name": "System Event Log Entry",  
    "Resolved": false,  
    "Severity": "OK"  
}
```

Setting Mode of 3-port Switch Ports to Allow Only BMC Port to Access OOB RJ45

```
{  
    "@odata.id":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/33",  
    "@odata.type": "#LogEntry.v1_13_0.LogEntry",  
    "AdditionalDataURI":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/33/att  
    "Created": "2024-06-11T13:20:09+00:00",  
    "EntryType": "Event",  
    "Id": "33",  
    "Message": "Only BMC port is allowed access to RJ45",  
    "Modified": "2024-06-11T13:20:09+00:00",  
    "Name": "System Event Log Entry",  
    "Resolved": false,  
    "Severity": "OK"
```

```
}
```

Clearing BMC SEL

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/2",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/2/attach",
  "Created": "2024-06-11T13:49:03+00:00",
  "EntryType": "Event",
  "Id": "2",
  "Message": "Start clearing SEL",
  "Modified": "2024-06-11T13:49:03+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

BMC Factory Reset

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/1",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/1/attach",
  "Created": "2024-06-11T13:49:03+00:00",
  "EntryType": "Event",
  "Id": "1",
}
```

```
    "Message": "BMC factory reset will take effect upon
reboot",
    "Modified": "2024-06-11T13:49:03+00:00",
    "Name": "System Event Log Entry",
    "Resolved": false,
    "Severity": "OK"
}
```

Resetting BMC Soft

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/17",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/17/att
  "Created": "2024-01-10T15:52:46+00:00",
  "EntryType": "Event",
  "Id": "17",
  "Message": "BMC Soft Reset",
  "Modified": "2024-01-10T15:52:46+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Enabling RShim Access from Host

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/3",
```

```
    "@odata.type": "#LogEntry.v1_13_0.LogEntry",
    "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/3/attach
    "Created": "2024-06-11T13:51:28+00:00",
    "EntryType": "Event",
    "Id": "3",
    "Message": "RShim access privilege from host will be
enabled after NIC reset",
    "Modified": "2024-06-11T13:51:28+00:00",
    "Name": "System Event Log Entry",
    "Resolved": false,
    "Severity": "OK"
}
```

Disabling RShim Access from Host

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/4",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/4/attach
  "Created": "2024-06-11T13:51:29+00:00",
  "EntryType": "Event",
  "Id": "4",
  "Message": "RShim access privilege from host will be
disabled after NIC reset",
  "Modified": "2024-06-11T13:51:29+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Enabling BlueField DPU Mode

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/31",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/31/att
  "Created": "2024-06-11T13:18:40+00:00",
  "EntryType": "Event",
  "Id": "31",
  "Message": "DPU mode will take effect after NIC reset",
  "Modified": "2024-06-11T13:18:40+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Enabling BlueField NIC Mode

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/30",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/30/att
  "Created": "2024-06-11T13:18:39+00:00",
  "EntryType": "Event",
  "Id": "30",
  "Message": "NIC mode will take effect after NIC reset",
  "Modified": "2024-06-11T13:18:39+00:00",
```

```
"Name": "System Event Log Entry",
"Resolved": false,
"Severity": "OK"
}
```

Enabling BlueField Secure Boot

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/28",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/28/att
  "Created": "2024-06-11T13:14:34+00:00",
  "EntryType": "Event",
  "Id": "28",
  "Message": "Secure Boot Option changed to Enable",
  "Modified": "2024-06-11T13:14:34+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Disabling BlueField Secure Boot

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/29",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/29/att
```

```
"Created": "2024-06-11T13:14:45+00:00",
"EntryType": "Event",
"Id": "29",
"Message": "Secure Boot Option changed to Disable",
"Modified": "2024-06-11T13:14:45+00:00",
"Name": "System Event Log Entry",
"Resolved": false,
"Severity": "OK"
}
```

Changing BlueField Boot Order

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/6",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/6/attach",
  "Created": "2024-06-11T13:02:04+00:00",
  "EntryType": "Event",
  "Id": "6",
  "Message": "System boot order changed",
  "Modified": "2024-06-11T13:02:04+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Enabling BlueField Boot Source

```
{
```

```
    "@odata.id":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/4",  
    "@odata.type": "#LogEntry.v1_13_0.LogEntry",  
    "AdditionalDataURI":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/4/attach",  
    "Created": "2024-07-26T09:49:42+00:00",  
    "EntryType": "Event",  
    "Id": "4",  
    "Message": "System boot source enabled",  
    "Modified": "2024-07-26T09:49:42+00:00",  
    "Name": "System Event Log Entry",  
    "Resolved": false,  
    "Severity": "OK"  
  }  
}
```

Disabling BlueField Boot Source

```
{  
  "@odata.id":  
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/5",  
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",  
  "AdditionalDataURI":  
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/5/attach",  
  "Created": "2024-07-26T09:49:42+00:00",  
  "EntryType": "Event",  
  "Id": "5",  
  "Message": "System boot source disabled",  
  "Modified": "2024-07-26T09:49:42+00:00",  
  "Name": "System Event Log Entry",  
  "Resolved": false,  
  "Severity": "OK"  
},
```

Changing BlueField Boot Source from Continuous to Once

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/3",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/3/attach",
  "Created": "2024-07-26T09:49:42+00:00",
  "EntryType": "Event",
  "Id": "3",
  "Message": "System boot source will take effect for one
boot",
  "Modified": "2024-07-26T09:49:42+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
},
```

Info

This log will not be generated if only the boot source is enabled without switching the boot override persistent setting

Changing BlueField Boot Source from Once to Continuous

```
{
```

```
    "@odata.id":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/3",  
    "@odata.type": "#LogEntry.v1_13_0.LogEntry",  
    "AdditionalDataURI":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/3/attach",  
    "Created": "2024-07-26T10:42:31+00:00",  
    "EntryType": "Event",  
    "Id": "3",  
    "Message": "System boot source will take effect  
continuously",  
    "Modified": "2024-07-26T10:42:31+00:00",  
    "Name": "System Event Log Entry",  
    "Resolved": false,  
    "Severity": "OK"  
}
```

Info

This log will not be generated if only the boot source is enabled without switching the boot override persistent setting

Changing BlueField Boot Source to Default

```
{  
    "@odata.id":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/11",  
    "@odata.type": "#LogEntry.v1_13_0.LogEntry",  
    "AdditionalDataURI":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/11/attach",  
    "Created": "2024-06-11T14:12:12+00:00",  
}
```

```
"EntryType": "Event",
"Id": "11",
"Message": "System boot source changed to Default",
"Modified": "2024-06-11T14:12:12+00:00",
"Name": "System Event Log Entry",
"Resolved": false,
"Severity": "OK"
}
```

Changing BlueField Boot Source to PXE

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/12",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/12/att
  "Created": "2024-06-11T14:12:13+00:00",
  "EntryType": "Event",
  "Id": "12",
  "Message": "System boot source changed to Network",
  "Modified": "2024-06-11T14:12:13+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Changing BlueField Boot Source to UEFI HTTP

```
{
```

```
    "@odata.id":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/12",  
    "@odata.type": "#LogEntry.v1_13_0.LogEntry",  
    "AdditionalDataURI":  
    "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/12/attachme  
    "Created": "2024-06-11T14:12:13+00:00",  
    "EntryType": "Event",  
    "Id": "12",  
    "Message": "System boot source changed to HTTP",  
    "Modified": "2024-06-11T14:12:13+00:00",  
    "Name": "System Event Log Entry",  
    "Resolved": false,  
    "Severity": "OK"  
  }
```

Changing BlueField Boot Type to Legacy

```
{  
  "@odata.id":  
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/9",  
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",  
  "AdditionalDataURI":  
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/9/attachme  
  "Created": "2024-06-11T14:09:40+00:00",  
  "EntryType": "Event",  
  "Id": "9",  
  "Message": "System boot type changed to Legacy",  
  "Modified": "2024-06-11T14:09:40+00:00",  
  "Name": "System Event Log Entry",  
  "Resolved": false,  
  "Severity": "OK"  
}
```

Changing BlueField Boot Type to UEFI

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/10",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/10/attach",
  "Created": "2024-06-11T14:10:43+00:00",
  "EntryType": "Event",
  "Id": "10",
  "Message": "System boot type changed to UEFI",
  "Modified": "2024-06-11T14:10:43+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Updating BlueField BFB Image

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/6",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/6/attach",
  "Created": "2024-06-11T14:01:13+00:00",
  "EntryType": "Event",
  "Id": "6",
  "Message": "Starting Bluefield DPU BFB update",
  "Modified": "2024-06-11T14:01:13+00:00",
}
```

```
"Name": "System Event Log Entry",
"Resolved": false,
"Severity": "OK"
}
```

Arm Frequency Change Redfish System Command

3 optional message descriptions:

- CPU frequency switched to P0 [100%].
- CPU frequency switched to P1 [80%].
- CPU frequency switched to P2 [50%].

Example:

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/5",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/5/attach",
  "Created": "2024-09-01T09:12:46+00:00",
  "EntryType": "Event",
  "Id": "5",
  "Message": "CPU frequency switched to P0 [100%].",
  "Modified": "2024-09-01T09:12:46+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "OK"
}
```

Data Port Module High Power Consumption Notification

An SEL entry generated when the power consumption of a data port module exceeds a critical threshold.

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/764",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/764/at-
  "Created": "2024-09-29T08:56:54+00:00",
  "EntryType": "Event",
  "Id": "764",
  "Message": "SEL event for port 1 High Module Current
notification, ThresholdCriticalHighGoingHigh",
  "Modified": "2024-09-29T08:56:54+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "Severity": "Critical"
}
```

Data Port Module Temperature Going High

Indicates that data port module temperature exceeded valid range.

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/SEL/Entries/5",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "Created": "2024-09-16T07:41:13+00:00",
  "EntryCode": "Assert",
  "EntryType": "SEL",
  "Id": "5",
```

```
    "Message": "SEL event for port 0 high thermal notification,
ThresholdWarningHighGoingHigh",
    "MessageId": "SEL event for port 0 high thermal
notification, ThresholdWarningHighGoingHigh",
    "Modified": "2024-09-16T07:41:13+00:00",
    "Name": "System Event Log Entry",
    "Resolved": false,
    "SensorNumber": 28,
    "SensorType": "Temperature",
    "Severity": "Warning"
}
```

Data Port Module Temperature Going Low

Indicates that data port module temperature returned to valid range.

```
{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/SEL/Entries/6",
  "@odata.type": "#LogEntry.v1_13_0.LogEntry",
  "Created": "2024-09-16T07:41:19+00:00",
  "EntryCode": "Assert",
  "EntryType": "SEL",
  "Id": "6",
  "Message": "SEL event for port 0 normal thermal
notification, ThresholdGoingLow",
  "MessageId": "SEL event for port 0 normal thermal
notification, ThresholdGoingLow",
  "Modified": "2024-09-16T07:41:19+00:00",
  "Name": "System Event Log Entry",
  "Resolved": false,
  "SensorNumber": 28,
  "SensorType": "Temperature",
  "Severity": "OK"
}
```

```
}
```

RAS Logging

CPER to Redfish severity translation:

CPER Severity	Redfish Severity
Recoverable	Warning
Fatal	Critical
Corrected	OK
Informational	Warning

RAS Cache Error

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/3",
  "@odata.type": "#LogEntry.v1_15_0.LogEntry",
  "AdditionalDataURI":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/3/attachments/1",
  "CPER": {
    "NotificationType": "09a9d5ac-5204-4214-96e5-
94992e752bcd",
    "Oem": {
      "Nvidia": {
        "@odata.type": "#NvidiaCPER.v1_0_0.NvidiaCPER",
        "ArmProcessor": {
          "ContextInfo": [],
          "ContextInfoNum": 0,
          "ErrorAffinity": {
            "Type": "Vendor Defined",
            "Value": 0
          }
        }
      }
    }
  }
}
```

```

    },
    "ErrorInfo": [
        {
            "CacheError": {
                "Corrected": false,
                "Level": 0,
                "Operation": {
                    "Name": "Generic Error",
                    "Value": 0
                },
                "PrecisePC": false,
                "ProcessorContextCorrupt": false,
                "RestartablePC": false,
                "TransactionType": {
                    "Name": "Instruction",
                    "Value": 0
                },
                "ValidationBits": {
                    "CorrectedValid": false,
                    "LevelValid": false,
                    "OperationValid": false,
                    "PrecisePCValid": false,
                    "ProcessorContextCorruptValid": false,
                    "RestartablePCValid": false,
                    "TransactionTypeValid": false
                }
            },
            "ErrorType": {
                "Name": "Cache Error",
                "Value": 0
            },
            "Flags": {
                "FirstErrorCaptured": false,
                "LastErrorCaptured": false,
                "Overflow": false,
            }
        }
    ]
}

```

```

        "Propagated": false
    },
    "Length": 32,
    "MultipleError": {
        "Type": "Multiple Errors",
        "Value": 1
    },
    "PhysicalFaultAddress": 0,
    "ValidationBits": {
        "ErrorInformationValid": false,
        "FlagsValid": false,
        "MultipleErrorValid": true,
        "PhysicalFaultAddressValid":
false,
        "VirtualFaultAddressValid": false
    },
    "Version": 0,
    "VirtualFaultAddress": 0
    }
],
"ErrorInfoNum": 1,
"MidrEl1": 1091556385,
"MpidrEl1": 2164326400,
"PsciState": 0,
"Running": true,
"SectionLength": 72,
"ValidationBits": {
    "ErrorAffinityLevelValid": false,
    "MpidrValid": true,
    "RunningStateValid": true,
    "VendorSpecificInfoValid": false
}
}
},
"SectionType": "e19e3d16-bc11-11e4-9caa-c2051d5d46b0"

```

```

    },
    "Created": "2024-11-15T19:14:48+00:00",
    "DiagnosticDataType": "CPERSection",
    "EntryType": "Event",
    "Id": "3",
    "Message": "A platform error occurred.",
    "MessageArgs": [],
    "MessageId": "Platform.1.0.PlatformError",
    "Name": "System Event Log Entry",
    "Resolution": "Check additional diagnostic data if
available.",
    "Resolved": false,
    "Severity": "Warning"
}

```

RAS Memory Error

```

{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/6",
  "@odata.type": "#LogEntry.v1_15_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/6/attach
  "CPER": {
    "NotificationType": "09a9d5ac-5204-4214-96e5-
94992e752bcd",
    "Oem": {
      "Nvidia": {
        "@odata.type": "#NvidiaCPER.v1_0_0.NvidiaCPER",
        "Memory": {
          "Bank": {
            "Value": 0
          },
        },
      },
    },
  },
}

```

```
"BitPosition": 0,
"Card": 0,
"CardSmbiosHandle": 0,
"Column": 0,
"Device": 0,
"ErrorStatus": {
  "AddressSignal": true,
  "ControlSignal": false,
  "DataSignal": false,
  "DetectedByRequester": false,
  "DetectedByResponder": false,
  "ErrorType": {
    "Description": "Storage error in
memory (DRAM).",
    "Name": "ERR_MEM",
    "Value": 4
  },
  "FirstError": false,
  "OverflowDroppedLogs": false
},
"Extended": {
  "ChipIdentification": 0,
  "RowBit16": false,
  "RowBit17": false
},
"MemoryErrorType": {
  "Name": "Scrub Uncorrected Error",
  "Value": 14
},
"ModuleRank": 0,
"ModuleSmbiosHandle": 0,
"Node": 0,
"PhysicalAddress": 12884901888,
"PhysicalAddressMask": 281474976710655,
"RankNumber": 0,
"RequestorID": 0,
```

```
"ResponderID": 0,
"Row": 40960,
"TargetID": 0,
"ValidationBits": {
  "BankAddressValid": false,
  "BankGroupValid": true,
  "BankValid": true,
  "BitPositionValid": true,
  "CardHandleValid": false,
  "CardValid": false,
  "ChipIdentificationValid": false,
  "ColumnValid": true,
  "DeviceValid": false,
  "ErrorStatusValid": true,
  "ExtendedRowBitsValid": true,
  "MemoryErrorTypeValid": true,
  "MemoryPlatformTargetValid": false,
  "ModuleHandleValid": false,
  "ModuleValid": true,
  "NodeValid": false,
  "PhysicalAddressMaskValid": true,
  "PhysicalAddressValid": true,
  "PlatformRequestorIDValid": false,
  "PlatformResponderIDValid": false,
  "RankNumberValid": true,
  "RowValid": true
}
}
},
"SectionType": "a5bc1114-6f64-4ede-b863-3e83ed7c83b1"
},
"Created": "2024-11-15T10:40:08+00:00",
"DiagnosticDataType": "CPERSection",
"EntryType": "Event",
"Id": "6",
```

```

"Message": "A platform error occurred.",
"MessageArgs": [],
"MessageId": "Platform.1.0.PlatformError",
"Name": "System Event Log Entry",
"Resolution": "Check additional diagnostic data if
available.",
"Resolved": false,
"Severity": "Warning"
}

```

RAS PCIe Error

```

{
  "@odata.id":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/15",
  "@odata.type": "#LogEntry.v1_15_0.LogEntry",
  "AdditionalDataURI":
"/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/15/attac
  "CPER": {
    "NotificationType": "cf93c01f-1a16-4dfc-b8bc-
9c4daf67c104",
    "Oem": {
      "Nvidia": {
        "@odata.type": "#NvidiaCPER.v1_0_0.NvidiaCPER",
        "Pcie": {
          "AerInfo": {
            "Capabilites_control": 0,
            "Capability_header": 0,
            "Correctable_error_mask": 0,
            "Correctable_error_status": 0,
            "Correctable_error_status_hex":
"0x00000000",

```

```

        "Data":
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        "Tlp_header_0": 0,
        "Tlp_header_1": 0,
        "Tlp_header_2": 0,
        "Tlp_header_3": 0,
        "Uncorrectable_error_mask": 0,
        "Uncorrectable_error_severity": 0,
        "Uncorrectable_error_status": 0,
        "Uncorrectable_error_status_hex":
"0x00000000"
    },
    "BridgeControlStatus": {
        "ControlRegister": 0,
        "SecondaryStatusRegister": 0
    },
    "CapabilityStructure": {
        "Data":
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    },
    "CommandStatus": {
        "CommandRegister": 0,
        "StatusRegister": 0
    },
    "DeviceID": {
        "ClassCode": 0,
        "DeviceID": 5555,
        "DeviceIDHex": "0x15B3",
        "DeviceNumber": 0,
        "FunctionNumber": 0,
        "PrimaryOrDeviceBusNumber": 0,
        "SecondaryBusNumber": 0,
        "SegmentNumber": 0,
        "SlotNumber": 0,
        "VendorID": 41692
    },

```

```

        "DeviceSerialNumber": 0,
        "PortType": {
            "Name": "Unknown",
            "Value": 16777216
        },
        "ValidationBits": {
            "AerInfoValid": false,
            "BridgeControlStatusValid": false,
            "CapabilityStructureStatusValid": false,
            "CommandStatusValid": false,
            "DeviceIDValid": false,
            "DeviceSerialNumberValid": false,
            "PortTypeValid": false,
            "VersionValid": false
        },
        "Version": {
            "Major": 0,
            "Minor": 0
        }
    }
},
    "SectionType": "d995e954-bbc1-430f-ad91-b44dcb3c6f35"
},
"Created": "2025-02-12T12:09:42+00:00",
"DiagnosticDataType": "CPERSection",
"EntryType": "Event",
"Id": "15",
"Message": "A platform error occurred.",
"MessageArgs": [],
"MessageId": "Platform.1.0.PlatformError",
"Name": "System Event Log Entry",
"Resolution": "Check additional diagnostic data if
available.",
"Resolved": false,
"Severity": "OK"

```

```
}
```

ATX Power Error

```
{
  "@odata.id":
  "/redfish/v1/Systems/Bluefield/LogServices/EventLog/Entries/3",
  "@odata.type": "#LogEntry.v1_15_0.LogEntry",
  "CPER": {
    "NotificationType": "6d5244f2-2712-11ec-bea7-
cb3fdb95c786",
    "Oem": {
      "Nvidia": {
        "@odata.type": "#NvidiaCPER.v1_0_0.NvidiaCPER",
        "Nvidia": {
          "ErrorInstance": 0,
          "ErrorType": 4,
          "InstanceBase": 0,
          "RegisterCount": 1,
          "Registers": [
            {
              "Address": 0,
              "Value": 1
            }
          ],
          "Severity": {
            "Code": 1,
            "Name": "Fatal"
          },
          "Signature": "NBU",
          "Socket": 0
        }
      }
    }
  }
}
```

```

    },
    "SectionType": "6d5244f2-2712-11ec-bea7-cb3fdb95c786"
  },
  "Created": "2025-01-16T08:37:32+00:00",
  "DiagnosticData":
"Q1BFUgAD/////wEAAQAAAAAAAAAD4AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  "DiagnosticDataType": "CPERSection",
  "EntryType": "Event",
  "Id": "3",
  "Links": {
    "OriginOfCondition": {
      "@odata.id":
"/redfish/v1/Systems/Bluefield/Processors/CPU_0"
    }
  },
  "Message": "A platform error occurred.",
  "MessageArgs": [],
  "MessageId": "Platform.1.0.PlatformError",
  "Name": "System Event Log Entry",
  "Resolution": "Check additional diagnostic data if
available.",
  "Resolved": false,
  "Severity": "Critical"
}

```

BMC Sensor Data

SDR Sensor List

The following is a list of the available sensors maintained by the BMC including their type and name.

Sensor Name	Sensor Type	Source	Description
p0_link	Discrete	IPMB	Uplink port 0 link status <ul style="list-style-type: none">0x100 – connection OK0x200 – connection error
p1_link	Discrete	IPMB	Uplink port 1 link status <ul style="list-style-type: none">0x100 – connection OK0x200 – connection error
bluefield_temp	Temperature	IPMB	NVIDIA® BlueField® temperature
p0_temp	Temperature	IPMB	Uplink port 0 SFP temperature
p1_temp	Temperature	IPMB	Uplink port 1 SFP temperature
ddr_temp	Temperature	IPMB	DDR temperature
rtc_voltage	Voltage	IPMB	RTC battery voltage
power_envelope	Power	IPMB	<ul style="list-style-type: none">This sensor indicates the maximum power consumption allowed for BlueField-3This sensor is incompatible with unsecured Linux
soc_power	Power	IPMB	<ul style="list-style-type: none">This sensor indicates the current power consumption of the BlueField-3 DPU

Sensor Name	Sensor Type	Source	Description
			<ul style="list-style-type: none"> This sensor is incompatible with unsecured Linux
power_envelope_deviation	Power	Synthesized Sensor	<p>Measures the deviation of soc_power sensor value from power_envelope sensor value.</p> $\text{power_envelope_deviation} = \text{soc_power} - \text{power_envelope}$ <ul style="list-style-type: none"> power_envelope The sensor value should be negative for normal conditions. If the sensor value is positive, then SoC power has exceeded the allowed power envelope. If the value of soc_power is NaN, then the power_envelope_deviation sensor value will also be NaN.
1V_BMC	Voltage	BMC ADC	
1_2V_BMC	Voltage	BMC ADC	
1_8V	Voltage	BMC ADC	
1_8V_BMC	Voltage	BMC ADC	
2_5V	Voltage	BMC ADC	
3_3V	Voltage	BMC ADC	
3_3V_RGM	Voltage	BMC ADC	
5V	Voltage	BMC ADC	
12V_ATX	Voltage	BMC ADC	Input power rail from ATX (provided from gold fingers in case of power off when ATX power is off)
12V_PCIE	Voltage	BMC ADC	Input power rail from gold fingers

Sensor Name	Sensor Type	Source	Description
DVDD	Voltage	BMC ADC	
HVDD	Voltage	BMC ADC	
VDD	Voltage	BMC ADC	
VDDQ	Voltage	BMC ADC	
VDD_CPU_L	Voltage	BMC ADC	
VDD_CPU_R	Voltage	BMC ADC	

Note

IPMB sourced sensors are supported when operating in DPU mode only.

Sensor Redfish Commands

Getting List of Support Sensors

BlueField sensors are stored within the Sensors schema under the Chassis schema. To retrieve the list of supported sensors, execute the following command:

```
curl -k -u root:'<password>' -H 'Content-Type: application/json'
-X GET https://<bmc_ip>/redfish/v1/Chassis/Card1/Sensors
```

The following is an example of the anticipated output:

```
{
  "@odata.id": "/redfish/v1/Chassis/Card1/Sensors",
  "@odata.type": "#SensorCollection.SensorCollection",
```

```
"Description": "Collection of Sensors for this Chassis",
"Members": [
  {
    "@odata.id":
"/redfish/v1/Chassis/Card1/Sensors/power_envelope"
  },
  {
    "@odata.id":
"/redfish/v1/Chassis/Card1/Sensors/power_envelope_deviation"
  },
  {
    "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/soc_power"
  },
  {
    "@odata.id":
"/redfish/v1/Chassis/Card1/Sensors/bluefield_temp"
  },
  {
    "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/ddr_temp"
  },
  {
    "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/p0_temp"
  },
  {
    "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/p1_temp"
  },
  {
    "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/12V_ATX"
  },
  {
    "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/12V_PCIE"
  },
  {
    "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/1V_BMC"
  },
  {

```

```
"@odata.id": "/redfish/v1/Chassis/Card1/Sensors/1_2V_BMC"
},
{
  "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/1_8V"
},
{
  "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/1_8V_BMC"
},
{
  "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/2_5V"
},
{
  "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/3_3V"
},
{
  "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/3_3V_RGM"
},
{
  "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/5V"
},
{
  "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/DVDD"
},
{
  "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/HVDD"
},
{
  "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/VDD"
},
{
  "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/VDDQ"
},
{
  "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/VDD_CPU_L"
},
{
```

```

        "@odata.id": "/redfish/v1/Chassis/Card1/Sensors/VDD_CPU_R"
      },
      {
        "@odata.id":
"/redfish/v1/Chassis/Card1/Sensors/rtc_voltage"
      }
    ],
    "Members@odata.count": 24,
    "Name": "Sensors"
  }

```

Getting Data for Specific Sensor

```

curl -k -u root:'<password>' -H 'Content-Type: application/json'
-X GET
https://<bmc_ip>/redfish/v1/Chassis/Card1/Sensors/<sensor_name>

```

The following is an example of a temperature sensor BlueField reading:

```

curl -k -u root:'<password>' -H 'Content-Type: application/json'
-X GET
https://<bmc_ip>/redfish/v1/Chassis/Card1/Sensors/bluefield_temp
{
  "@odata.id":
"/redfish/v1/Chassis/Card1/Sensors/bluefield_temp",
  "@odata.type": "#Sensor.v1_2_0.Sensor",
  "Id": "bluefield_temp",
  "Name": "bluefield temp",
  "Reading": 43.0,
  "ReadingRangeMax": 255.0,
  "ReadingRangeMin": 0.0,

```

```
"ReadingType": "Temperature",
"ReadingUnits": "Cel",
"RelatedItem": [
  {
    "@odata.id": "/redfish/v1/Systems/Bluefield"
  }
],
"Status": {
  "Conditions": [],
  "Health": "OK",
  "HealthRollup": "OK",
  "State": "Enabled"
},
"Thresholds": {
  "LowerCaution": {
    "Reading": 5.0
  },
  "LowerCritical": {
    "Reading": 0.0
  },
  "UpperCaution": {
    "Reading": 95.0
  },
  "UpperCritical": {
    "Reading": 105.0
  }
}
}
```

Configuring Sensor Thresholds

The following commands set the thresholds for sensors that support setting a threshold:

```
curl -k -u root:'<password>' -X PATCH
https://<bmc_ip>/redfish/v1/Chassis/Card1/Sensors/<sensor name>/
-d '{"Thresholds":{"<Threshold name>": {"Reading":<value>}}}'
```

The following is an example of how to set the upper critical threshold for the BlueField temperature sensor:

```
curl -k -u root:'<password>' -X PATCH
https://<bmc_ip>/redfish/v1/Chassis/Card1/Sensors/bluefield_temp
-d '{"Thresholds":{"UpperCritical": {"Reading":100}}}'
{
  "@Message.ExtendedInfo": [
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The request completed successfully.",
      "MessageArgs": [],
      "MessageId": "Base.1.15.0.Success",
      "MessageSeverity": "OK",
      "Resolution": "None"
    }
  ]
}
```

Sensor IPMI Commands

BMC software supports reading chassis sensor information using the IPMItool.

The following subsections list commands which allow reading SDR data.

Displaying Sensor Data

Displays sensor data repository entry readings and their status.

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN sdr list
```

Displaying Extended Sensor Data

Displays extended sensor information.

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN sdr elist
```

Displaying Sensors and Thresholds

Displays sensors and thresholds in a wide table format.

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN sensor  
list
```

Displaying Sensor Data Records Specified by Sensor ID

Displays sensor data records specified by sensor ID.

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN sdr get  
<name>
```

Displaying All Records from SDR Repository of Specific Type

Displays all records from the SDR repository of a specific type.

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN sdr type  
<type>
```

Displaying Data for Sensors Specified by Name

Displays information for sensors specified by name.

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN sensor  
get <sensor_name>
```

Displaying Readings for Sensors Specified by Name (Only for Numeric Sensors)

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U ADMIN -P ADMIN sensor  
reading <name>...<name>
```

BlueField Arm State

This section outlines methods for monitoring the state of NVIDIA® BlueField® Arm using either Redfish or IPMI.

Info

The BMC polls the host status from the NIC subsystem on BlueField using the NC-SI interface approximately every 30 seconds. Due to this implementation, some stages of the Arm boot process may not be captured. The expected `OemLastState` values to indicate boot completion for the different modes are as follows:

- `OsIsRunning` – for DPU mode
- `UEFI` – for NIC mode

Monitoring BlueField Arm State Using Redfish

```
curl -k -u root:<password> -H "Content-Type: application/json" -X GET  
https://<bmc_ip>/redfish/v1/Systems/Bluefield
```

The BlueField Arm state is represented by the `OemLastState` field under `BootProgress`.

Example output:

```
...  
"BootProgress": {
```

```
...  
  "OemLastState" : "OsIsRunning"  
}  
...
```

The possible values for `OemLastState` are:

- `BootRom`
- `BL2`
- `BL31`
- `UEFI`
- `OsStarting`
- `OsIsRunning`
- `LowPowerStandby`
- `FirmwareUpdateInProgress`
- `OsCrashDumpInProgress`
- `OsCrashDumpIsComplete`
- `FWFaultCrashDumpInProgress`
- `FWFaultCrashDumpIsComplete`
- `Invalid`

Monitoring BlueField Arm State Using IPMI

To get the BlueField Arm state with IPMI, refer to the `0xA3` command under "[IPMItool NIC Subsystem Management](#)".

Rsyslog

It is possible to dynamically configure rsyslog servers to receive system event log (SEL) messages and/or the BlueField SoC UART console printout (SOL) messages.

SEL and SOL Message Reception Format

SEL messages are received on the rsyslog server in the following format:

```
<Timestamp> <host> <EntryID-hex> | <Date> | <Time> | <Sensor-  
Type> | <Event-Type> | <Event-Direction> | <Description>
```

For example:

```
"2024-06-18T11:05:45.926095+03:00 ldev-platform-12-244.exam 75 |  
06/18/24 | 08:05:45 UTC | Voltage #0x08 | Lower Non-critical  
going low | Asserted"
```

SOL messages are received on the rsyslog server exactly as they appear in the BlueField console, including a timestamp and the hostname:

```
<Timestamp> <host> <message>
```

For example:

```
"2024-06-18T15:16:28.240538+03:00 ldev-platform-12-244  
systemd[1]: Starting RDMA Node Description Daemon"
```

Note

`$EscapeControlCharactersOnReceive` and `$Escape8BitCharactersOnReceive` should be turned off on the rsyslog server side.

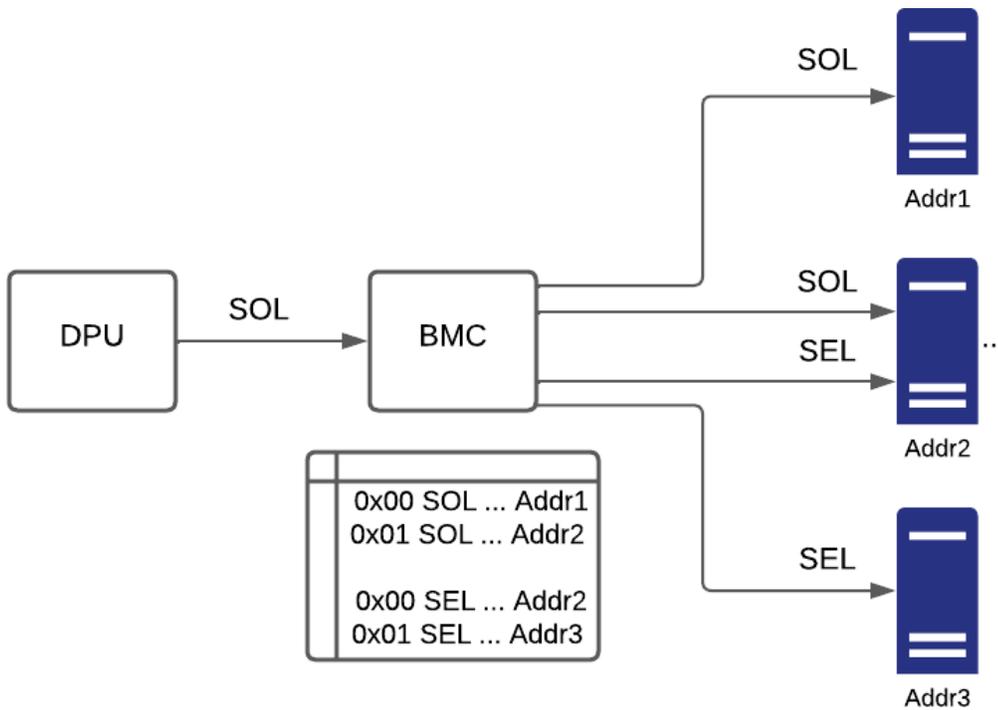
Rsyslog Servers Configurations

The rsyslog configurations define data streams. Each of them includes:

- Configuration identifier – An index (ranging from 0x00 to 0x09) AND a log type (SEL 0x01 or SOL 0x03)
- Status – Enable/disable
- Transport protocol – TCP/UDP
- Network protocol – IPv4/IPv6
- Server address
- Port

Note that configurations with the same index but different log types are considered to be different configurations. For example, 0x01-SOL and 0x01-SEL are distinct configurations.

The following diagram illustrates an example of three rsyslog servers receiving four data streams:



This setup requires four configurations:

- Configuration 0x00-SOL – Server1 receives SOL
- Configuration 0x01-SOL – Server2 receives SOL
- Configuration 0x00-SEL – Server2 receives SEL
- Configuration 0x01-SEL – Server3 receives SEL

Note

The BMC rsyslog configuration files located under `/etc/rsyslog.d` are automatically generated and are read-only. These files can only be modified using the IPMI commands listed later on this page.

IPMI Commands

The following table lists the IPMI commands for setting and getting rsyslog servers configurations:

netfunc	cmd	data	Description
0x32	0xD3	<Index> <LogType>	<p>Get rsyslog status – Displays information of the configured rsyslog server</p> <p>The request contains the index and the log type of the rsyslog server configuration, and it is 2 bytes long. The response contains the following information:</p> <pre> <Index> <LogType> <Status> <TransportProtocol> <NetworkProtocol> <ServerAddress> <Port> </pre> <ul style="list-style-type: none"> • Byte 1 – Completion code: <ul style="list-style-type: none"> ◦ 0x00 – Success (does not appear in IPMI textual response) ◦ 0x01 – Failure (the rest does not appear in IPMI response) • Byte 2 – Index <ul style="list-style-type: none"> ◦ Index of server (0x00-0x09) • Byte 3 – LogType <ul style="list-style-type: none"> ◦ 0x01 – SEL ◦ 0x03 – SOL • Byte 4 – Status <ul style="list-style-type: none"> ◦ 0x00 – Disabled ◦ 0x01 – Enabled • Byte 5 – Transport protocol <ul style="list-style-type: none"> ◦ 0x00 – UDP ◦ 0x01 – TCP • Byte 6 – Network protocol <ul style="list-style-type: none"> ◦ 0x00 – IPv4 ◦ 0x01 – IPv6 • Byte 7-n – Rsyslog server address <ul style="list-style-type: none"> ◦ Rsyslog addr (4/16 Bytes)

netfunc	cmd	data	Description
			<ul style="list-style-type: none"> Byte n+1-n+2 – Port <ul style="list-style-type: none"> Rsyslog port. LSB first. <p>The response is 12 bytes long for IPv4 and 24 bytes long for IPv6.</p>
0x32	0xD4	<pre><Index> <LogType> <Status> <TransportProtocol> <NetworkProtocol> <ServerAddress> <Port></pre>	<p>Set rsyslog status –</p> <ul style="list-style-type: none"> Configures a new rsyslog server configuration if the configuration <code><Index> <LogType></code> does not exist. Modifies an existing rsyslog server configuration if the configuration <code><Index> <LogType></code> does exist. <p>The command contains the following information:</p> <ul style="list-style-type: none"> Byte 1 – Index <ul style="list-style-type: none"> Index of server (0x00-0x09) Byte 2 – LogType <ul style="list-style-type: none"> 0x01 – SEL 0x03 – SOL Byte 3 – Status <ul style="list-style-type: none"> 0x00 – Disabled 0x01 – Enabled Byte 4 - Transport protocol <ul style="list-style-type: none"> 0x00 – UDP 0x01 – TCP Byte 5 - Network protocol <ul style="list-style-type: none"> 0x00 – IPv4 0x01 – IPv6 Byte 6-n – Rsyslog server address <ul style="list-style-type: none"> Rsyslog addr (4/16 Bytes) Byte n+1-n+2 – Port <ul style="list-style-type: none"> Rsyslog port. LSB first.

netfunc	cmd	data	Description
			The command data is 11 bytes long for IPv4 and 23 bytes long for IPv6. The response contains the completion code and is 1 byte long. The success completion code does not appear in IPMI textual response.

Usage Examples

Setting Rsyslog Status of Two Configurations

The following commands create or modify two different rsyslog configurations with Index 0x00 and LogTypes SEL/SOL :

netfunc: 0x32, cmd: 0xD4, Indx: 0x00, LogType: 0x01(SEL) / 0x03(SOL), status: 0x01 (Enabled), TP: 0x01 (TCP), NP: 0x00 (IPv4) Address: 0x0A 0xED 0x33 0xF4 (10.237.51.244) Port: 0xFA 0x13 (5114)

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD4 0x00 0x01 0x01 0x01 0x00
0x0A 0xED 0x33 0xF4 0xFA 0x13
root@dpu-bmc:~# ipmitool raw 0x32 0xD4 0x00 0x03 0x01 0x01 0x00
0x0A 0xED 0x33 0xF4 0xFA 0x13
```

Now the same rsyslog server receives both SEL and SOL messages.

The following command disables the rsyslog configurations with Index 0x00 and LogTypes SOL:

netfunc: 0x32, cmd: 0xD4, Indx: 0x00, LogType: 0x03 (SOL), status: 0x00 (Disabled), TP: 0x01 (TCP), NP: 0x00 (IPv4) Address: 0x0A 0xED 0x33 0xF4 (10.237.51.244) Port: 0xFA 0x13 (5114)

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD4 0x00 0x03 0x00 0x01 0x00
0x0A 0xED 0x33 0xF4 0xFA 0x13
```

Now the rsyslog server receives only SEL messages as the SOL configuration is disabled:

netfunc: 0x32, cmd: 0xD3, Indx: 0x00, LogType: 0x01(SEL) / 0x03(SOL)

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD3 0x00 0x01
00 01 01 01 00 0a ed 33 f4 fa 13
root@dpu-bmc:~# ipmitool raw 0x32 0xD3 0x00 0x03
00 03 00 01 00 0a ed 33 f4 fa 13
```

Setting Rsyslog Status with IPv6 Address

The following command creates or modified an rsyslog configuration with an IPv6 address:

netfunc: 0x32, cmd: 0xD4, Indx: 0x07, LogType: 0x01 (SEL), status: 0x01 (Enabled), TP: 0x01 (TCP), NP: 0x01 (IPv6) Address: 0xFD 0xFD 0xFD 0xFD 0x00 0x10 0x02 0x37 0x02 0x50 0x56 0xFF 0xFE 0x30 0x33 0xF4 (FDFD:FDFD:10:237:250:56FF:FE30:33F4) Port: 0xFA 0x13 (5114)

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD4 0x07 0x01 0x01 0x01 0x01
0xfd 0xfd 0xfd 0xfd 0x00 0x10 0x02 0x37 0x02 0x50 0x56 0xff 0xfe
0x30 0x33 0xf4 0xfa 0x13
```

Setting Rsyslog Status with Invalid Argument

The following command attempts to create an rsyslog server configuration with an invalid index 0x0A (Valid indexes are 0x00-0x09):

netfunc: 0x32, cmd: 0xD4, Indx: 0x0A, LogType: 0x01 (SEL), status: 0x01 (Enabled), TP: 0x01 (TCP), NP: 0x00 (IPv4) Address: 0x0A 0xED 0x33 0xF4 (10.237.51.244) Port: 0xFA 0x13 (5114)

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD4 0x0A 0x01 0x01 0x01 0x00
0x0A 0xED 0x33 0xF4 0xFA 0x13
```

```
Unable to send RAW command (channel=0x0 netfn=0x32 lun=0x0
cmd=0xd4 rsp=0xcc): Invalid data field in request
```

Getting Rsyslog Status Information

The following command displays the information of the rsyslog configuration with index 0 and LogType SEL :

```
netfunc: 0x32, cmd: 0xD3, Indx: 0x00, LogType: 0x01(SEL)
```

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD3 0x00 0x01
00 01 01 01 00 0a ed 33 f4 fa 13
```

Getting Non-existing Rsyslog Server Information

The following command attempts to receive an information of a non-existing rsyslog configuration with index 0x06 and LogType SEL :

```
netfunc: 0x32, cmd: 0xD3, Indx: 0x06, LogType: 0x01(SEL)
```

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD3 0x06 0x01
Unable to send RAW command (channel=0x0 netfn=0x32 lun=0x0
cmd=0xd3 rsp=0xcc): Invalid data field in request
```

DPU Chassis

The Redfish `Chassis` schema provides a structured and standardized way to represent essential information about the physical infrastructure of computing systems (i.e., the NVIDIA® BlueField®), offering valuable insights for system administrators, data center operators, and management software developers.

The BlueField chassis encompasses all system components, which include the `Bluefield_BMC`, `Bluefield_ERoT`, and `Card1` (which represents the BlueField).

```
curl -k -u root:'<password>' -H 'Content-Type: application/json'
-X GET https://<bmc_ip>/redfish/v1/Chassis
```

Output example:

```
{
  "@odata.id": "/redfish/v1/Chassis",
  "@odata.type": "#ChassisCollection.ChassisCollection",
  "Members": [
    {
      "@odata.id": "/redfish/v1/Chassis/Bluefield_BMC"
    },
    {
      "@odata.id": "/redfish/v1/Chassis/Bluefield_ERoT"
    },
    {
      "@odata.id": "/redfish/v1/Chassis/Card1"
    }
  ],
  "Members@odata.count": 3,
  "Name": "Chassis Collection"
```

```
}
```

Chassis Card1

```
curl -k -u root:'<password>' -H 'Content-Type: application/json'  
-X GET https://<bmc_ip>/redfish/v1/Chassis/Card1
```

Output example:

```
{  
  "@odata.id": "/redfish/v1/Chassis/Card1",  
  "@odata.type": "#Chassis.v1_21_0.Chassis",  
  "Actions": {  
    "#Chassis.Reset": {  
      "@Redfish.ActionInfo":  
"/redfish/v1/Chassis/Card1/ResetActionInfo",  
      "target": "/redfish/v1/Chassis/Card1/Actions/Chassis.Reset"  
    }  
  },  
  ..  
  "ChassisType": "Card",  
  "EnvironmentMetrics": {  
    "@odata.id": "/redfish/v1/Chassis/Card1/EnvironmentMetrics"  
  },  
  "Id": "Card1",  
  "Links": {  
    "ComputerSystems": [  
      {  
        "@odata.id": "/redfish/v1/Systems/Bluefield"  
      }  
    ],  
    "Contains": [  

```

```

    {
      "@odata.id": "/redfish/v1/Chassis/Bluefield_ERoT"
    },
    {
      "@odata.id": "/redfish/v1/Chassis/Bluefield_BMC"
    }
  ],
  "ManagedBy": [
    {
      "@odata.id": "/redfish/v1/Managers/Bluefield_BMC"
    }
  ]
},
"Manufacturer": "Nvidia",
"Model": "Bluefield 3 SmartNIC Main Card",
"Name": "Card1",
"NetworkAdapters": {
  "@odata.id": "/redfish/v1/Chassis/Card1/NetworkAdapters"
},
"PCIeDevices": {
  "@odata.id": "/redfish/v1/Chassis/Card1/PCIeDevices"
},
"PCIeSlots": {
  "@odata.id": "/redfish/v1/Chassis/Card1/PCIeSlots"
},
"PartNumber": "900-9D3B4-00EN-EAB",
"Power": {
  "@odata.id": "/redfish/v1/Chassis/Card1/Power"
},
"PowerState": "On",
"PowerSubsystem": {
  "@odata.id": "/redfish/v1/Chassis/Card1/PowerSubsystem"
},
"SKU": "",
"Sensors": {
  "@odata.id": "/redfish/v1/Chassis/Card1/Sensors"
}

```

```

    },
    "SerialNumber": "MT2245X00175",
    "Status": {
      "Conditions": [],
      "Health": "OK",
      "HealthRollup": "OK",
      "State": "Enabled"
    },
    "Thermal": {
      "@odata.id": "/redfish/v1/Chassis/Card1/Thermal"
    },
    "ThermalSubsystem": {
      "@odata.id": "/redfish/v1/Chassis/Card1/ThermalSubsystem"
    },
    "TrustedComponents": {
      "@odata.id": "/redfish/v1/Chassis/Card1/TrustedComponents"
    },
    "UUID": ""
  }
}

```

Chassis Card1 NetworkAdapters

Note

Retrieving these values is supported when operating in DPU mode only.

The `NetworkAdapters` schema specifically aims to standardize NIC management and representation. This schema includes a collection under `NvidiaNetworkAdapter` where each element holds the following fields:

- `Ports`

The following is an example of the network port associated with `eth0`. Note that the naming conventions may differ depending on your device configuration.

```
curl -k -u root:'PASSWORD' -H 'Content-Type:
application/json' -X GET
https://<IP>/redfish/v1/Chassis/Card1/NetworkAdapters/NvidiaNe
```

Example output:

```
{
  "@odata.id":
  "/redfish/v1/Chassis/Card1/NetworkAdapters/NvidiaNetworkAdapte
  "@odata.type": "#Port.v1_6_0.Port",
  "CurrentSpeedGbps": 200,
  "Id": "eth0",
  "LinkNetworkTechnology": "Ethernet",
  "LinkStatus": "LinkUp",
  "Name": "Port"
}
```

- `NetworkDeviceFunctions`

The following is an example of the network device function for `eth0f0` (i.e., `eth0` function 0). Note that the naming conventions may differ depending on your device configuration.

```
curl -k -u root:'PASSWORD' -H 'Content-Type:
application/json' -X GET
https://<IP>/redfish/v1/Chassis/Card1/NetworkAdapters/NvidiaNe
```

Example output:

```

{
  "@odata.id":
"/redfish/v1/Chassis/Card1/NetworkAdapters/NvidiaNetworkAdapter/NetworkDeviceFunctions/eth0f0"
  "@odata.type": "#NetworkDeviceFunction.v1_9_0.NetworkDeviceFunction",
  "Ethernet": {
    "MACAddress": "02:8e:00:2d:4f:f8",
    "MTUSize": 1500
  },
  "Id": "eth0f0",
  "Links": {
    "OffloadSystem": {
      "@odata.id": "/redfish/v1/Systems/Bluefield"
    },
    "PhysicalPortAssignment": {
      "@odata.id":
"/redfish/v1/Chassis/Card1/NetworkAdapters/NvidiaNetworkAdapter/Ports/eth0"
    }
  },
  "Name": "NetworkDeviceFunction",
  "NetDevFuncCapabilities": [
    "Ethernet"
  ],
  "NetDevFuncType": "Ethernet"
}

```

Note

Removing or adding new ports requires a BMC reboot.

DPU Information

Note

The following actions are supported when operating in DPU mode only.

Getting Base GUID

```
curl -k -u root:'<password>' -X GET  
https://<bmc_ip>/redfish/v1/redfish/v1/Systems/Bluefield/Oem/Nvidia | jq  
' .BaseGUID'
```

Getting Base MAC

```
curl -k -u root:'<password>' -X GET  
https://<bmc_ip>/redfish/v1/redfish/v1/Systems/Bluefield/Oem/Nvidia | jq  
' .BaseMAC'
```

Getting Description

```
curl -k -u root:'<password>' -X GET  
https://<bmc_ip>/redfish/v1/redfish/v1/Systems/Bluefield/Oem/Nvidia | jq  
' .Description'
```

BMC and BlueField Logs

The BMC and NVIDIA® BlueField® logs can be collected using Redfish commands.

Two types of dumps are supported:

- BMC dump, which is a collection of logs from BMC
- System dump, which is a collection of logs from BlueField. To create a system dump, users must provide the BlueField credentials and IP address of the `tmfifo_net0` network interface.

BMC Dump Operations

The following subsections list BMC dump operations.

Creating BMC Dump Task

Create a BMC dump task and gets the task ID.

Info

This is important for the next stages.

```
sudo curl -k -u root:'<password>' -d '{"DiagnosticDataType":  
"Manager"}' -X POST  
https://<ip_address>/redfish/v1/Managers/Bluefield_BMC/LogServices.
```

Where:

- `<ip-address>` – BMC IP address

- `<password>` – BMC password

(i) Note

This command triggers an attempt to enable the RShim on the BMC.

(i) Note

Notes about the size of a single BMC dump and the BMC dumps container:

- The total size of all BMC dumps cannot exceed 8MB
- A single BMC dump cannot take up more than 4MB. If it is larger, it is truncated to 4MB.
- For the proper creation of a BMC dump, 4MB of free memory are required regardless of its actual size (can be smaller than 4MB). This memory is ensured by deleting existing BMC dumps, from oldest to newest, until 4MB are free.

Getting Dump Task State

Get dump task state. When `TaskState` is `Completed`, then the dump is ready for download.

```
sudo curl -k -u root:'<password>' -H 'Content-Type: application/json' -X GET https://<ip_address>/redfish/v1/TaskService/Tasks/<task_id>
```

Where:

- `<ip-address>` – BMC IP address
- `<password>` – BMC password
- `<task_id>` – task ID received from the first command

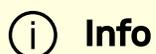
Downloading BMC Dump

Download BMC dump after `TaskState` is `Completed`. Dump is saved in the path given to `--output`.

```
sudo curl -k -u root:'<password>' -H 'Content-Type: application/json' -X GET https://<ip_address>/redfish/v1/Managers/Bluefield_BMC/LogServices, --output </path/to/tar/log_dump.tar.xz>
```

Where:

- `<ip-address>` – BMC IP address
- `<password>` – BMC password
- `<entry_id>` – entry ID of the dump in `redfish/v1/Managers/Bluefield_BMC/LogServices/Dump/Entries/`
- `</path/to/tar/log_dump.tar.xz>` – path to download the log dump `log_dump.tar.xz`



Info

After downloading, untar the file to view the logs.

Log list:

- `journal-pretty.log`
- `sensor-readings.log`
- `host-state.log`
- `hostnamectl.log`
- `fw-version.log`
- `fru-info.log`
- `nicDeviceDebugInfo.log (mstdump)`
 - CRspace and Scartchpad address spaces
 - Dumps are lists of 32-bit addresses and the 32-bit values stored at these addresses
 - Only works if NIC is working
 - Only on BlueField-3
- `chassis-state.log`
- `bmc-state.log`
- `rshim.log`
- `uptime.log`
- `cpuinfo`
- `fw-printenv.log`

- `varfilelist.log`
- `tmpfilelist.log`
- `softIRQs.log`
- `sensorinfo.log`
- `selinfo.log`
- `pslist.log`
- `routeinfo.log`
- `network.log`
- `network`
 - `00-bmc-eth0.network`
 - `00-tmfifo_net0.network`
 - `00-bmc-vlan4040.network`
 - `vlan4040.netdev`
- `netstat.log`
- `mntinfo.log`
- `kernalcmdline.log`
- `kernalRingBuff.log`
- `iproute.log`
- `dpulogs`
 - `dpu_console`

- `obmc-console.log`
- `iplink.log`
- `ipaddr.log`
- `interrupts.log`
- `freemem.log`
- `channelconfig.log`
- `channelaccess.log`
- `arptable.log`
- `inventory.log`
- `elogall.log`
- `os-release`
- `top.log`
- `meminfo`
- `failed-services.log`
- `hwmon.log`
- `tmpfilelist.log`
- `slabinfo.log`
- `settings.log`
- `dmesg.log`
- `em-system.json`

- `dmesg.log`
- `bios.log`
- `timedate.log`
- `procfld.log`
- `dreport.log`
- `disk-usage.log`
- `summary.log`

i Note

To access logs under `bflogs`, BlueField would have to operate in DPU mode.

Deleting All Dump Entries

Clear all log dump entries.

```
sudo curl -k -u root:'<password>' -H 'Content-Type: application/json' -X POST https://<ip_address>/redfish/v1/Managers/Bluefield_BMC/LogServices,
```

Where:

- `<ip-address>` – BMC IP address
- `<password>` – BMC password

Specific log dump entry deletion can be done by using 'curl's DELETE instead of GET in the previous command.

System Dump Operations

The following subsections list system dump operations.

Creating System Dump

Create a system dump and get task ID.

```
sudo curl -k -u root:'<password>' -d '{"DiagnosticDataType":  
"OEM", "OEMDiagnosticDataType": "bf_ip=<bf_ip>;bf_username=  
<bf_username>;bf_password=<bf_password>"}' -X POST  
https://<ip_address>/redfish/v1/Systems/Bluefield/LogServices/Dump,
```

Where:

- `<ip-address>` – BMC IP address
- `<password>` – BMC password
- `<bf_ip>` – BlueField IP address
- `<bf_username>` – BlueField username
- `<bf_password>` – BlueField password

Info

Note, this command triggers an attempt to enable the RShim on the BMC.

Getting Dump Task State

Get dump task state. The dump is ready for download when `TaskState` is `Completed`.

```
sudo curl -k -u root:'<password>' -H 'Content-Type:
application/json' -X GET
https://<ip_address>/redfish/v1/TaskService/Tasks/<task_id>
```

Where:

- `<ip-address>` – BMC IP address
- `<password>` – BMC password
- `<task_id>` – task ID received from the first command

Downloading System Dump

Download the user-specified system dump.

```
sudo curl -k -u root:'<password>' -H 'Content-Type:
application/json' -X GET
https://<ip_address>/redfish/v1/Systems/Bluefield/LogServices/Dump,
--output </path/to/tar/system_dump.tar.xz>
```

Where:

- `<ip-address>` – BMC IP address
- `<password>` – BMC password

- `<entry_id>` – The entry ID of the dump can be found in `redfish/v1/Managers/Bluefield_BMC/LogServices/Dump/Entries/`
- `</path/to/tar/system_dump.tar.xz>` – path to download the log dump `system_dump.tar.xz`

i Info

After downloading, untar the file to view the logs.

Dump list:

- `bflogs/dmesg`
- `bflogs/lastlog`
- `bflogs/wtmp`

i Note

To access logs under `bflogs`, BlueField would have to operate in DPU mode.

- `rshim.log`
- `dreport.log`
- `summary.log`

Deleting All Dump Entries

Clear all log dump entries.

```
sudo curl -k -u root:'<password>' -H 'Content-Type: application/json' -X POST https://<ip_address>/redfish/v1/Systems/Bluefield/LogServices/Dump,
```

Where:

- `<ip-address>` – BMC IP address
- `<password>` – BMC password

Info

Specific log dump entry deletion can be done by using curl's DELETE instead of GET in the previous command.

The downloaded dump tar must be extracted to get the logs for BMC or BlueField.

Upon creating a dump, please allow the system ~5 mins to prepare the dump. The created dump will appear on the dump list when the system finishes dump creation. The created dump can be downloaded from the BMC using the `retrieve` command.

BlueField Console Log

BMC captures the BlueField console output and stores it in the System dump. Refer to section "[System Dump Operations](#)" for getting the log files in BMC dump.

Users may also check the log in `/run/log/dpu/logs/`. The log is rotated if it is larger than 1M or older than 24 hours. The oldest console output is overwritten as new data is added.

System Processor

Note

System processor information is sourced from the UEFI and stored in the BMC's persistent memory. This data may be lost after a BMC factory reset and will be restored upon the next UEFI reboot.

Processor Summary

A high-level summary of the BlueField processors is available in the `ProcessorSummary` object within the Redfish `ComputerSystem` schema.

To retrieve this information, run the following command:

```
curl -k -u <bmc_username>:<bmc_password> -H 'Content-Type: application/json' -X GET https://<bmc_ip>/redfish/v1/Systems/Bluefield
```

Example output:

```
{
  ...,
  "@odata.id": "/redfish/v1/Systems/Bluefield",
  "@odata.type": "#ComputerSystem.v1_22_0.ComputerSystem",
  ...,
  "ProcessorSummary": {
    "CoreCount": 16,
    "Count": 1,
```

```
        "Model": "ARMv8"
    },
    "Processors": {
        "@odata.id": "/redfish/v1/Systems/Bluefield/Processors"
    },
    ...
}
```

The `ProcessorSummary` provides:

- `CoreCount` – Total number of cores across all processors.
- `Count` – Total number of processors.
- `Model` – Processor model of the primary processor.

Processor Collection

A detailed list of system processors is available in the `ProcessorCollection` object.

To retrieve the processor collection, run:

```
curl -k -u <bmc_username>:<bmc_password> -H 'Content-Type:
application/json' -X GET
https://<bmc_ip>/redfish/v1/Systems/Bluefield/Processors
```

Example output:

```
{
  "@odata.id": "/redfish/v1/Systems/Bluefield/Processors",
  "@odata.type": "#ProcessorCollection.ProcessorCollection",
  "Members": [
    {
```

```
        "@odata.id" :
"/redfish/v1/Systems/Bluefield/Processors/CPU_0"
    }
],
"Members@odata.count": 1,
"Name": "Processor Collection"
}
```

Each entry in the `Members` array represents a processor in the system. In this example, the collection contains a single processor: `CPU_0`.

Individual Processor Information

Detailed information about an individual processor is provided in the Redfish `Processor` schema.

To retrieve information for `CPU_0`, run the following command:

```
curl -k -u <bmc_username>:<bmc_password> -H 'Content-Type:
application/json' -X GET
https://<bmc_ip>/redfish/v1/Systems/Bluefield/Processors/CPU_0
```

Example output:

```
{
  "@Redfish.Settings": {
    "@odata.type": "#Settings.v1_3_3.Settings",
    "SettingsObject": {
      "@odata.id":
"/redfish/v1/Systems/Bluefield/Processors/CPU_0/Settings"
    }
  },
}
```

```

    "@odata.id":
"/redfish/v1/Systems/Bluefield/Processors/CPU_0",
    "@odata.type": "#Processor.v1_20_0.Processor",
    "EnvironmentMetrics": {
        "@odata.id":
"/redfish/v1/Systems/Bluefield/Processors/CPU_0/EnvironmentMetrics
    },
    "Id": "CPU_0",
    "Links": {
        "Chassis": {
            "@odata.id": "/redfish/v1/Chassis/Card1"
        }
    },
    "Location": {
        "PartLocation": {
            "ServiceLabel": "Socket 0"
        }
    },
    "Manufacturer": "https://www.mellanox.com",
    "MaxSpeedMHz": 2135,
    "Metrics": {
        "@odata.id":
"/redfish/v1/Systems/Bluefield/Processors/CPU_0/ProcessorMetrics"
    },
    "Model": "Mellanox BlueField-3 [A1] A78(D42) 16 Cores r0p1",
    "Name": "Processor",
    "PartNumber": "OPN: 9009D3B600CVAA",
    "Ports": {
        "@odata.id":
"/redfish/v1/Systems/Bluefield/Processors/CPU_0/Ports"
    },
    "ProcessorId": {
        "EffectiveFamily": "0x0101",
        "IdentificationRegisters": "0x00000000410FD421"
    },
    "ProcessorType": "CPU",

```

```

"SerialNumber": "Unspecified Serial Number",
"Socket": "Socket 0",
"Status": {
  "Conditions": [],
  "Health": "OK",
  "State": "Enabled"
},
"TotalCores": 16,
"TotalThreads": 16,
"Version": "Mellanox BlueField-3 [A1] A78(D42) 16 Cores r0p1"
}

```

Supported Properties

Property	Type	Description
Id	string	Unique identifier of the processor.
MaxSpeedMHz	integer (MHz)	Maximum clock speed of the processor.
Model	string	Processor model number (matches the Version property).
Name	string	Name of the processor.
PartNumber	string	Part number assigned to the processor.
ProcessorType	string	Type of processor (e.g., CPU, GPU).
Status	object	Status object reporting health and operational state.
TotalCores	integer	Total number of cores in the processor.
TotalThreads	integer	Total number of execution threads supported by the processor.
Version	string	Hardware version of the processor.

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. NVIDIA Corporation ("NVIDIA") makes no 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 NON-INFRINGEMENT, 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 and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright 2025. PDF Generated on 12/15/2025