NVIDIA BlueField BMC Software v24.07

Table of contents

About This Document

BMC software enables control and management of the baseboard management controller's (BMC) hardware components. The BMC software supports the Intelligent Platform Management Interface (IPMI).

This guide provides general information concerning the BMC on the NVIDIA® BlueField® networking device (DPU or SuperNIC) and is intended for those who want to familiarize themselves with the functionality provided by the BMC.

Note

This document is relevant for BlueField devices with an integrated BMC. Please refer to the Supported Platforms and Interoperability page to ascertain whether your device features an integrated BMC.

Tip

If this is your first time deploying the BlueField device (Day 0), please refer to the "NVIDIA BlueField Management and Initial Provisioning Guide" for a walkthrough of the provisioning process. Afterwards, this manual is useful for BlueField lifecycle management (Day 1) activities.

Software Download

To download product software, please refer to the BlueField software product page.

Technical Support

Customers who purchased NVIDIA products directly from NVIDIA are invited to contact us through the following methods:

- E-mail: enterprisesupport@nvidia.com
- Enterprise Support page: https://www.nvidia.com/en-us/support/enterprise

Customers who purchased NVIDIA M-1 Global Support Services, please see your contract for details regarding technical support.

Customers who purchased NVIDIA products through an NVIDIA-approved reseller should first seek assistance through their reseller.

Related Documentation

Document Name	Description
NVIDIA BlueField DPU BSP	This document provides product release notes as well as information on the BlueField software distribution and how to develop and/or customize applications, system software, and file system images for the BlueField platform
NVIDIA BlueField-2 Ethernet DPU User Guide	This manual describes BlueField-2 Ethernet device including details as to the interfaces of the board, specifications, required software and firmware, and a step-by-step plan of how to bring it up
NVIDIA BlueField-3 Ethernet DPU User Guide	This manual describes BlueField-3 Ethernet device including details as to the interfaces of the board, specifications, required software and firmware, and a step-by-step plan of how to bring it up
BlueField DPU Administrator Quick Start Guide	This quick start guide details the procedure for installing a brand-new NVIDIA® BlueField®
NVIDIA BlueField Management	This document defines the NVIDIA-recommended method to manage NVIDIA® BlueField®-2 and BlueField®-3 devices, reviews BlueField

Document Name	Description
and Initial Provisioning	management interfaces, protocols, and capabilities (hardware, firmware, etc.), and explains how to use them to manage the BlueField.
Redfish Data Model Specification	This document includes the informative and normative descriptions copied from the description and long description annotations in the Redfish Schema Bundle (DSP8010), and adds supplemental normative text to further explain the usage of particular properties or resources.
IPMI Architecture GitHub	This document describes the architecture of IPMI design.

Glossary

Abbreviation / Acronym	Whole Word / Description
BMC	Baseboard management controller
DPU	Data processing unit
EEPROM	Electrically Erasable Programmable Read Only Memory
FRU	Field Replaceable Unit
IPMB	Intelligent Platform Management Bus
IPMI	Intelligent Platform Management Interface
NIC	Network interface card
SoC	System-on-chip
SOL	Serial Over LAN
SEL	System Event Log
SDR	Sensor Data Record; Sensor Data Repository
UART	Universal Asynchronous Receiver Transmitter

Release Notes

The following pages provide information on the supported platforms, changes and new features, and reports on software known issues as well as bug fixes.

Overview

The BMC node enables remote power cycling, board environment monitoring, NVIDIA® BlueField® SoC temperature monitoring, board power and consumption monitoring, and individual interface resets. The BMC also supports the ability to push a bootstream to the BlueField. It is recommended to manage NVIDIA® BlueField® using Redfish commands. However, IPMI commands and sysfs monitoring infrastructure are available as well .

> ⚠ **Warning**
>
> Make sure to log into the BMC first and change the global default password to prevent malicious attackers from hacking your system.

The procedures described in this manual assume that you have already installed and powered on your device according to the instructions in BlueField's specific hardware guide.

- Support for IPMI 2.0 (v1.1) Standards
    - Thermal control – access to all relevant temperature sensors, fan control
    - System management – power state control, power on/off, reboot/reset
    - Environmental monitoring – voltage/current/power
    - Serial over LAN (SOL)
    - RMCP/RMCP+
    - Event log management
    - Event alerting
    - VLAN support

- Support for DMTF Standards
    - Redfish specification (DSP0266)
    - Network Controller Sideband Interface (NC-SI) Specification (DMTF DSP0222)
- Support for BMC image update

Connecting to BMC Interfaces

BMC Management Interface

The BMC has a separate Ethernet interface which provides network connection for management traffic to the BMC. The NVIDIA® BlueField® networking platform's bracket (DPU or SuperNIC) has an RJ45 port labeled "MGMT" which is the management interface port. The management port is configured with auto-negotiation capabilities by default (100MbE to 1GbE).

The BMC interface `eth0` is the management interface, so any information displayed by `ifconfig eth0` pertains to the management interface. The MAC address to be used for `eth0` is pre-programmed in the BMC FRU EEPROM and can be found on the BlueField's board label. By default, the IP address used for `eth0` is acquired via DHCP but can be configured differently.

Changing Default Password

When initially logging into the system, it is mandatory to update the default BMC password, `0penBmc`. The BlueField BMC offers two methods/interfaces for changing the password:

- SSH/serial:

  To change the password, connect to the BMC via SSH/serial and log in using the root user and the default password. Upon logging in, you are prompted with the following:

  ```
  dpu-bmc login: root
  Password: <Type default password>
  You are obliged to immediately change your password (mandatory for administrators).
  Changing the root password.
  Current password: <Retype the default password>
  ```

```
New password: <Type the new password according to the above rules>
Retype the new password: <Retype the new password>
```

- Redfish:

  The Redfish user management interface may be used to configure the new password. The following Redfish command can be employed to alter the default password:

```
curl -k -u root:0penBmc -H "Content-Type: application/json" -X PATCH
https://<bmc_ip>/redfish/v1/AccountService/Accounts/root -d '{"Password" : "<password>"}'
```

The BMC password must comply with the following policy parameters:

- Using ASCII and Unicode characters is permitted
- Minimum length: 12
- Maximum length: 20
- Maximum number of consecutive character pairs: 4

> **Info**
>
> Two characters are consecutive if `|hex(char_1)-hex(char_2)|=1`.
>
> Examples of passwords with 5 consecutive character pairs (invalid): `DcBa123456AbCd!`; `ab1XbcYcdZdeGef!`; `Testing_123abcgh!`.

The following is a valid example password:

- HelloNvidia3D!

> **Note**

A user account is locked for 10 minutes after 10 consecutive failed attempts.

Account Service

The Redfish root user can inquire about and modify the applied account policies, which encompass settings such as the number of consecutive login attempts permitted and the time period for which the system will remain locked.

The following Redfish command provides the current settings:

```
curl -k -u root:'<password>' -H 'Content-Type: application/json' -X GET
https://10.237.53.58/redfish/v1/AccountService
```

Example output:

```
{
  "@odata.id": "/redfish/v1/AccountService",
  "@odata.type": "#AccountService.v1_10_0.AccountService",
  "AccountLockoutDuration": 600,
  "AccountLockoutThreshold": 4,
  "Accounts": {
    "@odata.id": "/redfish/v1/AccountService/Accounts"
  },
  "ActiveDirectory": {
    "Authentication": {
      "AuthenticationType": "UsernameAndPassword",
      "Password": null,
      "Username": ""
    },
    "LDAPService": {
      "SearchSettings": {
        "BaseDistinguishedNames": [
          ""
        ],
```

```
      "GroupsAttribute": "",
      "UsernameAttribute": ""
    }
  },
  "RemoteRoleMapping": [],
  "ServiceAddresses": [
    ""
  ],
  "ServiceEnabled": false
},
"Description": "Account Service",
"Id": "AccountService",
"LDAP": {
  "Authentication": {
    "AuthenticationType": "UsernameAndPassword",
    "Password": null,
    "Username": ""
  },
  "Certificates": {
    "@odata.id": "/redfish/v1/AccountService/LDAP/Certificates"
  },
  "LDAPService": {
    "SearchSettings": {
      "BaseDistinguishedNames": [
        ""
      ],
      "GroupsAttribute": "",
      "UsernameAttribute": ""
    }
  },
  "RemoteRoleMapping": [],
  "ServiceAddresses": [
    ""
  ],
  "ServiceEnabled": false
},
"MaxPasswordLength": 20,
"MinPasswordLength": 13,
"Name": "Account Service",
"Oem": {
  "OpenBMC": {
    "@odata.id": "/redfish/v1/AccountService#/Oem/OpenBMC",
    "@odata.type": "#OemAccountService.v1_0_0.AccountService",
    "AuthMethods": {
```

```
      "BasicAuth": true,
      "Cookie": true,
      "SessionToken": true,
      "TLS": true,
      "XToken": true
    }
   }
  },
  "Roles": {
   "@odata.id": "/redfish/v1/AccountService/Roles"
  },
  "ServiceEnabled": true
}
```

By default, if a user attempts to log into the system with an incorrect password four times in a row, their account is locked for 600 seconds. Afterwards, the user is allowed another opportunity to log in with the correct credentials. If the user fails to log in again, the account is immediately locked for an additional 600 seconds. If the user logs in successfully, the counter of consecutive login failures is reset.

The `patch` command may be used to modify the default policy settings. The following example illustrates how to alter the number of allowed consecutive login attempts into the system.

```
curl -k -u root:'<password>' -H 'Content-Type: application/json' -X PATCH
https://<IP>/redfish/v1/AccountService -d '{"AccountLockoutThreshold" : 10}'
```

Info

For a comprehensive understanding of the schema, please refer to the DMTF definition of the AccountService.v1_10_0.AccountService schema.

If an account becomes inaccessible, users may check the system's status using the Redfish interface using the following GET operation:

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

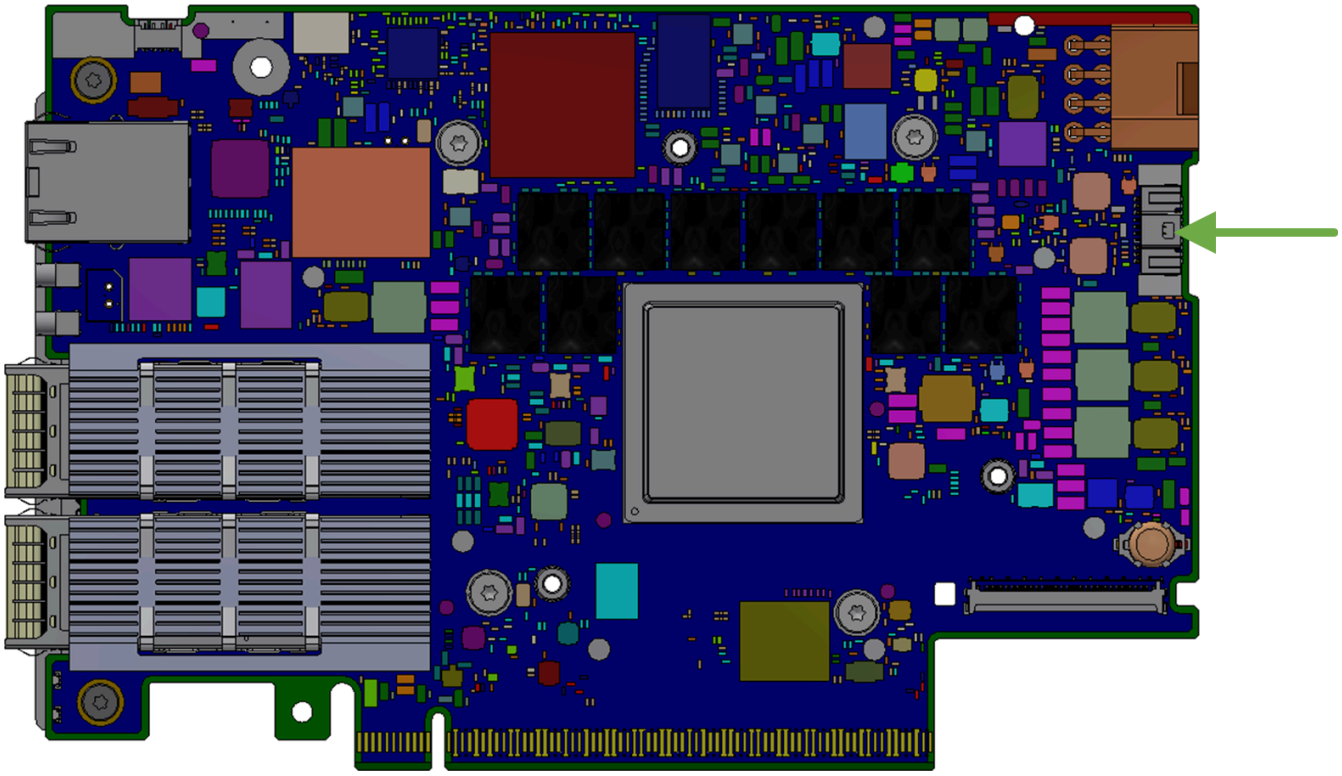
Example output:

```
{
  "error": {
    "@Message.ExtendedInfo": [
      {
        "@odata.type": "#Message.v1_1_1.Message",
        "Message": "While accessing the resource at '/redfish/v1/AccountService', the service received an
authorization error 'Account temporarily locked out for 600 seconds due to multiple authentication
failures'.",
        "MessageArgs": [
          "/redfish/v1/AccountService",
          "Account temporarily locked out for 600 seconds due to multiple authentication failures"
        ],
        "MessageId": "Base.1.15.0.ResourceAtUriUnauthorized",
        "MessageSeverity": "Critical",
        "Resolution": "Ensure that the appropriate access is provided for the service in order for it to
access the URI."
      }
    ],
    "code": "Base.1.15.0.ResourceAtUriUnauthorized",
    "message": "While accessing the resource at '/redfish/v1/AccountService', the service received an
authorization error 'Account temporarily locked out for 600 seconds due to multiple authentication
failures'."
  }
}
```

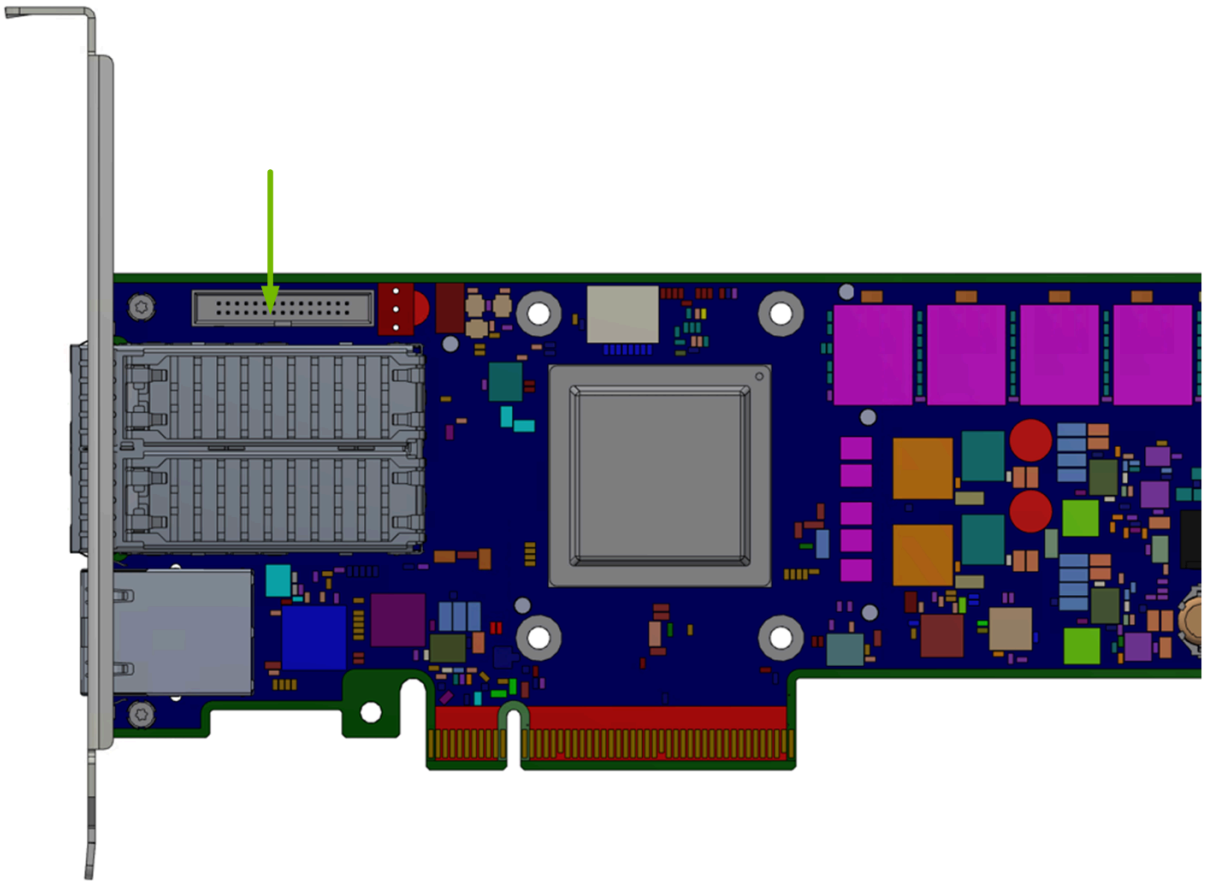
BMC Console Interface

The BMC UART1 console is available on the IO panel. The BMC is connected to a 20-pin connector for BlueField-3 or 30-pin connector for BlueField-2 which allows the Linux console to be monitored.

BlueField-3 BMC Connector



BlueField-2 BMC Connector



Network Configuration

> **Warning**
>
> Do not manually modify the network configuration file /etc/systemd/network/00-bmc-eth0.network.

There are two ways of configuring the network interfaces:

- Dynamic (DHCP)
- Static

See section "Network Protocol Support" for more details.

BMC USB Port

This section describes the use cases for the BMC USB port. Note that only BMC Linux has access to the USB port and its feature set. There is no access to BMC USB port while running u-boot.

> **Note**
>
> Due to a hardware bug in AST2500, the USB interface is only able to work at USB 1.0 speeds.

> **Note**
>
> Storage device support on this port has only been validated with USB flash drives.

Providing Removable Storage via USB Stick

Once a USB stick is plugged in to the BMC's USB port, issue the command `lsusb` and/or check the dmesg log to see if the USB stick has been detected. The successful insertion of a USB stick will create a device under `/dev` called `sda` (or `sdb`), and a mountable partition `/dev/sda1`. To mount the USB stick as a filesystem, just issue the command "`mount /dev/sda1 /mnt`" to mount it at `/mnt`. The command "`umount /mnt`" unmounts the device.

Platform Management Interface

NVIDIA® BlueField® provides management interfaces to the BMC and the BlueField device.

Redfish Management Interface

The BlueField's BMC provides a standard DMTF Redfish management interface, which is accessible via an HTTPS RESTful interface. This Redfish interface enables users to inquire about and configure the system:

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

Redfish Implementation for NVIDIA BlueField BMC

Redfish Specification DSP0266 defines the necessary protocols, data model, behaviors, and other architectural components for creating an interoperable, multivendor, remote, and out-of-band capable interface. This interface is designed to meet the scalable platform management expectations of cloud-based and web-based IT professionals.

The specification outlines the mandatory elements required for all Redfish implementations and the optional elements that system vendors and manufacturers can choose to include. It also specifies where implementations can provide OEM-specific extensions.

NVIDIA BlueField BMC's Redfish implementation is based on the bmcweb implementation from the OpenBmc open-source community, ensuring compliance with the Redfish specification. As part of the BlueField development cycle, the code is synchronized with the upstream community, making it subject to changes and modifications derived from the upstream implementation. While NVIDIA is committed to adhering to the Redfish specification, the actual implementation may vary due to this development process.

Redfish POST (Action)

The bmcweb POST operation implementation complies with the DSP0266 specification definition. For more information, refer to the "POST (action)" section of Redfish Specification DSP0266.

IPMI

The BMC, based on the Intelligent Platform Management Interface (IPMI) standard, supports both out-of-band (OOB) dedicated interfaces, and a serial port to access the CLI of the BMC.

External Host Retrieving Data from BMC Via UART

If an external host is connected and logged into the BMC via UART, IPMI commands can be issued to fetch information from the BMC as follows:

```
ipmitool <ipmitool_arguments>
```

External Host Retrieving Data from BMC Via LAN

The BMC is connected to an external host server via LAN. IPMItool commands may be issued from the external server to retrieve information from the BMC as follows:

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

BMC Management

NVIDIA BMC is based on the OpenBMC open-software framework which builds a complete Linux image for a board management controller (BMC). It uses the Yocto project as the underlying building and distro generation framework.

The primary software components of BMC are the following:

- U-boot bootloader
- Linux kernel
- OpenBMC distro

This section consists of the following pages:

- Software Versioning
- Product Instance Identifier
- Boot Sequence Overview
- User Management
- Network Protocol Support
- LLDP in Redfish
- Reset or Reboot BMC
- Factory Reset BMC
- CEC and BMC Firmware Operations
- BlueField BMC Redfish Triggers
- Redfish Certificate Management

- Guest Tunnel

System Management

This section contains the following pages:

Appendix – Software Upgrade Provisioning Flow

This appendix details the steps for provisioning software components on NVIDIA® BlueField®-3 networking platform (DPU or SuperNIC).

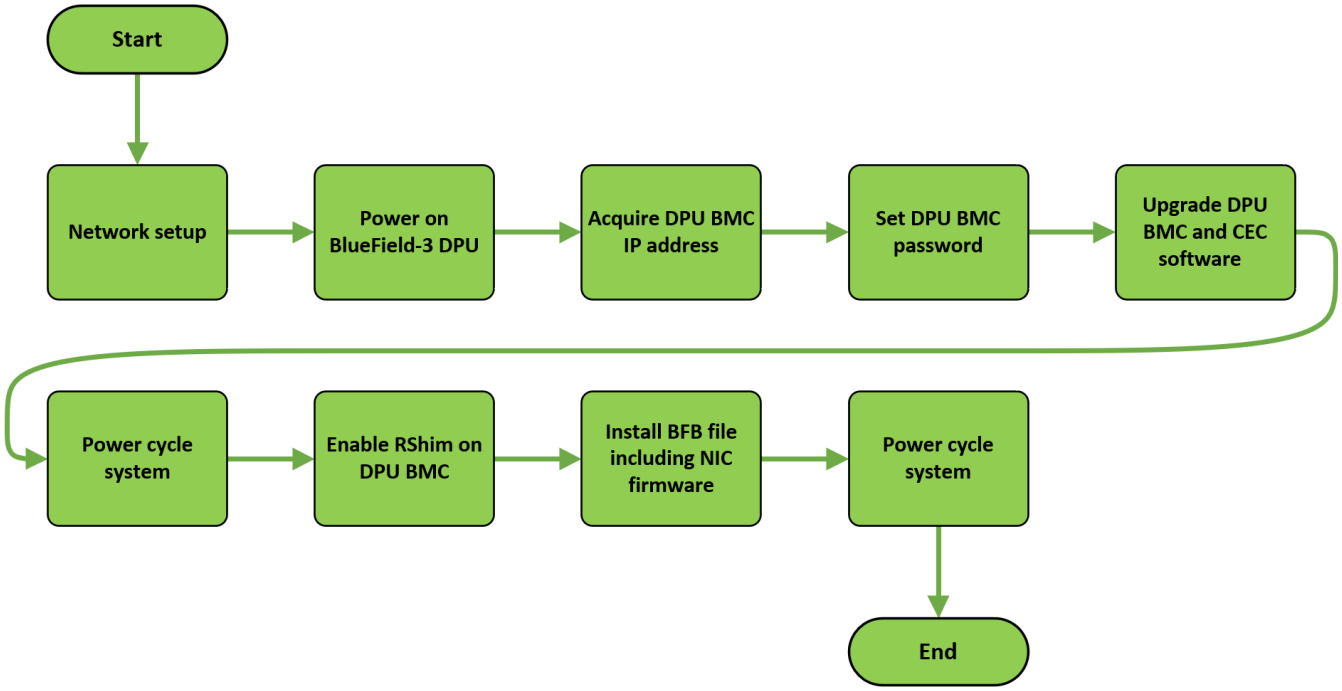
> **Info**
>
> The procedure for BlueField BMC software upgrade is agnostic to the version of the software. Once upgraded, however, the procedure assumes you to be running the latest BMC software.

This workflow guarantees the most current software to be installed on various components of BlueField-3. This includes:

- BlueField BMC
- CEC
- Arm ATF
- Arm UEFI
- Arm OS
- NIC firmware

The process aims to ensure that all these components are up to date.

The following high-level flow diagram outlines the expected steps to be followed throughout the process:



1. Establish a connection between the onboard RJ-45 network interface and the management network. Refer to section "Network Protocol Support" for detailed instructions on network connectivity.

2. Power on the BlueField device. This can be accomplished manually or by utilizing either ipmitool or Redfish commands directed at the host's BMC.

    - IPMItool example:

      ```
      ipmitool -H <bmc_ip_or_hostname> -U <username> -P <password> power on
      ```

      Replace the parameters with the information relevant for your host BMC.

    - Redfish example:

      ```
      curl -X POST -k -u root:<password> -H "Content-Type: application/json" -d '{"ResetType": "On"}' https://<bmc_ip>/redfish/v1/Systems/<System_ID>/Actions/ComputerSystem.Reset
      ```

      Replace the parameters with the information relevant for your host BMC.

3. Acquire the BlueField BMC's MAC address from the label affixed to the BlueField (highlighted in the image). Use the BlueField BMC's MAC address to retrieve the assigned IP address from the DHCP server to enable communication with the BlueField BMC over the network.

4. If BlueField-3 is a new device which has not yet been provisioned, the BlueField BMC comes from the factory with a default password (0penBmc). To establish communication with the BlueField BMC, you must change the default password. Refer to section "Changing Default Password" for instructions on changing the default password of the BlueField BMC.

5. Upgrade BlueField BMC and CEC software. This step is crucial for guaranteeing that all new features and functionalities are available on your device. Refer to section "BMC and CEC Firmware Update" for instructions on how to do that.

6. Power cycle the host. This can be accomplished by utilizing either ipmitool or Redfish commands directed at the host's BMC:

    1. IPMItool example:

        ```
        ipmitool -H <bmc_ip_or_hostname> -U <username> -P <password> power cycle
        ```

        Replace the parameters with the information relevant for your host BMC.

2. Redfish example:

```
curl -k -u root:<password> -X POST
"https://<host_bmc_ip>/redfish/v1/Systems/1/Actions/ComputerSystem.Reset" -d
'{"ResetType": "ForceRestart"}'
```

   Replace the parameters with the information relevant for your host BMC.

7. Ensure that the RShim is disconnected from the host to enable the BlueField BMC to take ownership of it. To achieve this, follow the following steps in section "Enabling RShim on BMC" under "Installing BFB".

8. Install the BFB file and NIC firmware.

```
# echo WITH_NIC_FW_UPDATE=yes > bf.cfg
# cat <path_to_bfb> bf.cfg > new.bfb
```

   Follow the instructions provided in the BFB image transfer guidelines provided in section "Transferring BFB Image" under "Installing BFB" while utilizing the newly created BFB file, new.bfb.

9. To ensure that the new NIC firmware takes effect, perform a final power cycle of the system as detailed in step 6.

List of Supported IPMItool Commands

The IPMItool program allows you to remotely manage the IPMI functions of the NVIDIA® BlueField® BMC. The commands below may be directed to the BMC's Ethernet interface by invoking:

```
ipmitool -C 17 -I lanplus -H <bmc_ip_addr> -U ADMIN -P ADMIN <ipmitool_arguments>
```

The following list provides a full list of the IPMItool arguments supported by BlueField BMC.

```
chassis power reset
chassis status (to be implemented in future release)
fru
fru print 0
fru print 1
fru read 0 /tmp/fru
fru read 1 /tmp/fru
lan print
mc info
mc reset cold
sdr elist
sdr get <sensor name>
sdr list
sdr type <type>
sel
sel clear
sel elist
sel listsensor get <sensor name>
sensor list
sol activate
user disable <user id>
```

```
user enable <user id>
user list [<channel number>]
user priv <user id> <privilege level(1-4)> [<channel number>]
user set name <user id> <user name>
user set password <user id> <password>
```

NVIDIA OEM Commands

Not all functionalities are covered with a standard set of IPMItool commands. Therefore, a set of custom NVIDIA IPMItool raw commands have been added. The first two parameters of the raw command are NetFN and CMD.

IPMItool raw commands follow the following format:

```
ipmitool -C 17 -I lanplus -H <bmc_ip> -U <username> -P <password> raw <netfunc> <cmd> <data>
```

Where:

- netfunc – network function which identifies the functional message class, and clusters IPMI commands into sets
- cmd – one byte command within a network function
- data – optional element which provides additional parameters for a request or response message

netfunc	cmd	data	Description
0x32	0x66	N/A	Factory reset
0x32	0x67	0x00	Disable vendor field mode settings to be run from Arm OS
0x32	0x67	0x01	Enable vendor field mode settings to be run from Arm OS
0x32	0x68	N/A	Fetch vendor field mode settings to be run from Arm OS
0x32	0x6a	0	Stops RShim on BMC

netfunc	cmd	data	Description
0x32	0x6a	1	Starts RShim on BMC
0x32	0x69	N/A	Retrieves RShim service status on BMC. Expected output: • 0x00 – RShim inactive (default state) • 0x01 – RShim active
0x32	0x6b	N/A	Gets the DNS server
0x32	0x6c	0x0b 0x31 0x30 0x2e 0x31 0x35 0x2e 0x31 0x32 0x2e 0x36 0x37	Adds the DNS server
0x32	0x92	N/A	Enters NVIDIA® BlueField® into Livefish (FNP) mode
0x32	0x93	N/A	Disable Livefish (FNP) mode
0x32	0xa1	0x0	OEM command 0xa1 is defined for various reset controls of BlueField from BMC under the OEM NetFn group 0x30. • 0x00 – hard reset of BlueField
0x32	0xa7	N/A	Gets NTP servers.
0x32	0xa8 0x00	0x31 0x2e 0x69 0x6e 0x2e 0x70 0x6f 0x6f 0x6c 0x2e 0x6e 0x74 0x70 0x2e 0x6f 0x72 0x67	Adds primary NTP server with address 1.in.pool.ntp.org
0x32	0xa8 0x01	0x31 0x2e 0x69 0x6e 0x2e 0x70 0x6f 0x6f 0x6c 0x2e 0x6e 0x74 0x70 0x2e 0x6f 0x72 0x67	Adds secondary NTP server with address 1.in.pool.ntp.org **Note** Should be added after a primary server has been

netfunc	cmd	data	Description
			declared.
0x32	0xa8 0x02	0x01	Enable time sync to NTP server
0x32	0xa8 0x02	0x00	Disables NTP time sync
0x32	0xD2	N/A	Triggers a CEC self reset for CEC pending firmware activation

NIC Subsystem Management

> **Note**
>
> This content is relevant for NVIDIA® BlueField®-3 devices only.

Redfish NIC Subsystem Management

Configuring BlueField Mode of Operation

Refer to "BlueField Modes of Operation Configuration" for information.

Getting Host RShim

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

Enabling Host RShim

```
curl -k -u root:'<password>' -H "Content-Type: application/json" -X POST -d '{"HostRshim":"Enabled"}'
https://<bmc_ip>/redfish/v1/Systems/Bluefield/Oem/Nvidia/Actions/HostRshim.Set
```

Disabling Host RShim

```
curl -k -u root:'<password>' -H "Content-Type: application/json" -X POST -d '{"HostRshim":"Disabled"}'
https://<bmc_ip>/redfish/v1/Systems/Bluefield/Oem/Nvidia/Actions/HostRshim.Set
```

Getting Strap Options

```
curl -k -u root:'<password>' -X GET
https://<bmc_ip>/redfish/v1/Systems/Bluefield/Oem/Nvidia/Connectx/StrapOptions
```

Getting External Host Privileges

```
curl -k -u root:'<password>' -X GET
https://<bmc_ip>/redfish/v1/Systems/Bluefield/Oem/Nvidia/Connectx/ExternalHostPrivileges
```

Setting External Host Privileges

```
curl -k -u root:'<password>' -H "Content-Type: application/json" -X POST -d
'{"HostPrivFwUpdate":"Default","HostPrivNvGlobal":"Enabled" ...}'
https://<bmc_ip>/redfish/v1/Systems/Bluefield/Oem/Nvidia/Connectx/ExternalHostPrivileges/Actions/Exte
```

- Currently, firmware does not support the parameters HOST_PRIV_FLASH_ACCESS and HOST_PRIV_PCC_UPDATE. Their value should stay as DEVICE_DEFAULT.
- The parameter HOST_PRIV_NV_INTERNAL_CPU should either equal the parameter HOST_PRIV_NV_GLOBAL or one of them should be set to DEVICE_DEFAULT.
- If the parameter HOST_PRIV_FLASH_ACCESS is not set to DEVICE_DEFAULT then the following parameters should all be set to DEVICE_DEFAULT or be equal to the value of

HOST_PRIV_FLASH_ACCESS: HOST_PRIV_NV_HOST, HOST_PRIV_NV_PORT, HOST_PRIV_NV_GLOBAL, HOST_PRIV_NV_INTERNAL_CPU, HOST_PRIV_PCC_UPDATE, HOST_PRIV_FW_UPDATE.

IPMItool NIC Subsystem Management

Since the standard IPMItool commands do not cover all functionality, a set of custom NVIDIA IPMItool raw commands is available to enable configuring the NIC subsystem on the BlueField directly.

IPMItool raw commands follow the following format:

```
ipmitool -C 17 -I lanplus -H <bmc_ip_addr> -U <username> -P <password> raw <netfunc> <cmd> <data>
```

Where:

- netfunc – network function which identifies the functional message class, and clusters IPMI commands into sets
- cmd – one byte command within a network function
- data – optional element which provides additional parameters for a request or response message

The following table lists the supported IPMItool raw commands:

<table>
<tr><th>netfunc</th><th>cmd</th><th>data</th><th>Description</th></tr>
<tr><td>0x32</td><td>0x9A</td><td>N/A</td><td>Get external host privileges.
Prints current state for all fields:
<table><tr><th>Byte</th><th>Field</th></tr><tr><td>0</td><td>HOST_PRIV_FLASH_ACCESS</td></tr></table></td></tr>
</table>

<table>
<tr><th>netfunc</th><th>cmd</th><th>data</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Byte</th><th>Field</th></tr>
<tr><td>1</td><td>HOST_PRIV_FW_UPDATE</td></tr>
<tr><td>2</td><td>HOST_PRIV_NIC_RESET</td></tr>
<tr><td>3</td><td>HOST_PRIV_NV_GLOBAL</td></tr>
<tr><td>4</td><td>HOST_PRIV_NV_HOST</td></tr>
<tr><td>5</td><td>HOST_PRIV_NV_INTERNAL_CPU</td></tr>
<tr><td>6</td><td>HOST_PRIV_NV_PORT</td></tr>
<tr><td>7</td><td>HOST_PRIV_PCC_UPDATE</td></tr>
</table>
Each state is represented by binary byte in order.
• 00 – Default
• 01 – Enabled
• 02 – Disabled
</td></tr>
<tr><td>0x32</td><td>0x9B</td><td>Byte0
Byte1</td><td>Set external host privilege.
Byte0 selects privilege according to the following table:
<table>
<tr><th>Byte</th><th>Field</th></tr>
<tr><td>0</td><td>HOST_PRIV_FLASH_ACCESS</td></tr>
<tr><td>1</td><td>HOST_PRIV_FW_UPDATE</td></tr>
<tr><td>2</td><td>HOST_PRIV_NIC_RESET</td></tr>
<tr><td>3</td><td>HOST_PRIV_NV_GLOBAL</td></tr>
<tr><td>4</td><td>HOST_PRIV_NV_HOST</td></tr>
<tr><td>5</td><td>HOST_PRIV_NV_INTERNAL_CPU</td></tr>
<tr><td>6</td><td>HOST_PRIV_NV_PORT</td></tr>
<tr><td>7</td><td>HOST_PRIV_PCC_UPDATE</td></tr>
</table>
</td></tr>
</table>

netfunc	cmd	data	Description
			Byte1 is the value being set. Supported values: • 00 – Default • 01 – Enabled • 02 – Disabled **Note** • Currently, firmware does not support the parameters HOST_PRIV_FLASH_ACCESS and HOST_PRIV_PCC_UPDATE. Their value should stay as DEVICE_DEFAULT. • The parameter HOST_PRIV_NV_INTERNAL_CPU should either equal the parameter HOST_PRIV_NV_GLOBAL or one of them should be set to DEVICE_DEFAULT. • If the parameter HOST_PRIV_FLASH_ACCESS is not set to DEVICE_DEFAULT then the following parameters should all be set to DEVICE_DEFAULT or be equal to the value of HOST_PRIV_FLASH_ACCESS: HOST_PRIV_NV_HOST, HOST_PRIV_NV_PORT, HOST_PRIV_NV_GLOBAL, HOST_PRIV_NV_INTERNAL_CPU, HOST_PRIV_PCC_UPDATE, HOST_PRIV_FW_UPDATE.
0x32	0x9C	N/A	Get SmartNIC mode. Prints current configuration: INTERNAL_CPU_OFFLOAD_ENGINE. • 00 – Disabled • 01 – Enabled

<table>
<tr><th>netfunc</th><th>cmd</th><th>data</th><th>Description</th></tr>
<tr><td>0x32</td><td>0x9D</td><td>Byte0</td><td>Set SmartNIC mode (INTERNAL_CPU_OFFLOAD_ENGINE) to Byte0.
Supported values:
00 – Disabled
01 – Enabled</td></tr>
<tr><td>0x32</td><td>0x9E</td><td>N/A</td><td>Get host access. Prints current HOST_PRIV_RSHIM.
00 – Disabled
01 – Enabled</td></tr>
<tr><td>0x32</td><td>0x9F</td><td>Byte0</td><td>Set host access. Sets HOST_PRIV_RSHIM to Byte0.
Supported values:
00 – Disabled
01 – Enabled</td></tr>
<tr><td>0x32</td><td>0xA2</td><td>N/A</td><td>Query strap options. Prints current state for all fields:<table><tr><th>Byte</th><th>Field</th></tr><tr><td>0</td><td>VERSION</td></tr><tr><td>1</td><td>DISABLE_INBAND_RECOVER_VALUE</td></tr><tr><td>2</td><td>PRIMARY_IS_PCORE_1_VALUE</td></tr><tr><td>3</td><td>2PCORE_ACTIVE_VALUE</td></tr><tr><td>4</td><td>SOCKET_DIRECT_VALUE</td></tr><tr><td>5</td><td>PCI_REVERSAL_VALUE</td></tr><tr><td>6</td><td>PCI_PARTITION_1_VALUE</td></tr><tr><td>7</td><td>PCI_PARTITION_0_VALUE</td></tr><tr><td>8</td><td>OSC_FREQ_1_VALUE</td></tr><tr><td>9</td><td>OSC_FREQ_0_VALUE</td></tr></table></td></tr>
</table>

<table>
<tr><th>netfunc</th><th>cmd</th><th>data</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Byte</th><th>Field</th></tr>
<tr><td>10</td><td>CORE_BYPASS_N_VALUE</td></tr>
<tr><td>11</td><td>FNP_VALUE</td></tr>
<tr><td>12</td><td>DISABLE_INBAND_RECOVER_VALUE</td></tr>
<tr><td>13</td><td>PRIMARY_IS_PCORE_1_MASK</td></tr>
<tr><td>14</td><td>2PCORE_ACTIVE_MASK</td></tr>
<tr><td>15</td><td>SOCKET_DIRECT_MASK</td></tr>
<tr><td>16</td><td>PCI_REVERSAL_MASK</td></tr>
<tr><td>17</td><td>PCI_PARTITION_1_MASK</td></tr>
<tr><td>18</td><td>PCI_PARTITION_0_MASK</td></tr>
<tr><td>19</td><td>OSC_FREQ_1_MASK</td></tr>
<tr><td>20</td><td>OSC_FREQ_0_MASK</td></tr>
<tr><td>21</td><td>CORE_BYPASS_N_MASK</td></tr>
<tr><td>22</td><td>FNP_MASK</td></tr>
</table>
Each state is represented by binary byte in order.
Supported values:

00 – Disabled
01 – Enabled

</td></tr>
<tr><td>0x32</td><td>0xA3</td><td>N/A</td><td>Get SmartNIC OS State.

00 – BootRom
01 – BL2
02 – BL31
03 – UEFI
04 – OsStarting
05 – OsIsRunning

</td></tr>
</table>

netfunc	cmd	data	Description
			• 06 – LowPowerStandby • 07 – FirmwareUpdateInProgress • 08 – OsCrashDumpInProgress • 09 – OsCrashDumpIsComplete • 0A – FWFaultCrashDumpInProgress • 0B – FWFaultCrashDumpIsComplete • 0C – Invalid

Setting Operation Mode

netfunc	cmd	data	Description
0x32	0x9D	0x1	Set DPU mode
0x32	0x9D	0x0	Set NIC mode

Enabling/Disabling RShim from Host

netfunc	cmd	data	Description
0x32	0x9F	0x1	Enable RShim from host
0x32	0x9F	0x0	Disable RShim from host

Table of Common Commands

Capability	Redfish	IPMItool	Supported in NIC Mode? [Y/N]
Changing the default BMC password	Changing default password using Redfish	N/A	Yes
Changing the default UEFI password	Changing UEFI Password	N/A	Yes
Enabling/disabling secure boot	Setting Secure Boot State	N/A	N/A
Updating BMC firmware	BMC and CEC firmware update	N/A	Yes
Updating BlueField BFB	Pushing BFB from BMC to BlueField Arm	N/A	Yes
Configuring BlueField to network boot from the out-of-band interface first	Boot Config Using Redfish	Boot Config Using IPMI	N/A
Resetting BlueField	Reset control	Reset control	Partial
Resetting BlueField BMC	Reset control using Redfish	Reset control using IPMI	Yes
Factory reset	Factory Reset Redfish Command	Factory Reset IPMI Command	Yes
Getting BlueField versions	System	N/A	No

Capability	Redfish	IPMItool	Supported in NIC Mode? [Y/N]
	inventory		
Getting BlueField BMC versions	Retrieving BMC version using Redfish	Retrieving BMC version using IPMI command	Yes
Getting high-speed ports MAC addresses for mapping BlueField's Ethernet devices	Chassis Card1 NetworkAdapters	List of IPMI Supported FRUs	No
BlueField monitoring (SEL, FRU, etc.)	Monitoring	Monitoring	No
User management	User management Redfish commands	User management IPMI commands	Yes
Enabling secure boot with customer keys	BIOS secure boot configuration	N/A	N/A
Enabling/disabling zero-trust mode	Disable host RShim	Enable/disable RShim from Host	Yes – BlueField-3 only
Enabling RShim from BlueField BMC	Enable RShim on BlueField BMC	Enable RShim	Yes
Changing BlueField mode	Redfish NIC Subsystem Management	Setting operation mode	Yes – BlueField-3 only
Partial BFB update (ATF/UEFI)	Deploying BlueField Software Using BFB from BMC	NA	Yes

Document Revision History

Rev v24.07 – August 14, 2024

Added:

- Page "Guest Tunnel"
- Page "Rsyslog"
- Section "Initial Provisioning of Golden Image to BMC"
- Section "Retrieving Golden Image Version Information"
- Section "Retrieving Golden Image Version Information Using Redfish"

Updated:

- Section "Redfish Management Interface"
- Diagram on page "CEC and BMC Firmware Operations"
- Multipart update with MultipartHttpPushUri command in section "Trigger Secure Firmware Update"
- Section "ForceUpdate"
- Section "Updating BMC"
- Section "Updating CEC"
- Section "Installing BFB"
- Section "Changing Default UEFI Password Using Redfish"
- Section "Retrieve Information on Pending Settings"
- Section "FRU Reading IPMI Commands"

- Section "Hard Rebooting BlueField"
- Section "Updating BMC"
- Section "RAS Errors"
- Section "System Commands"
- Page "BlueField Host Network Interface"

Rev v24.04 – May 05, 2024

Added:

- Page "BlueField Host Network Interface"
- Page "LLDP in Redfish"
- Page "Monitoring BlueField Arm State"
- Section "Product Instance Identifier"
- Section "CEC Activation and Reset" under "BMC and CEC Firmware Update"
- Section "BMC DPU Information"

Updated:

- Section "Installing BFB"
- Section "FRU Reading IPMI Commands"

Rev v24.01 – February 08, 2024

Added:

- Section "Monitoring DPU OS Shutdown from BMC"
- Section "Example for CSR Generation, Certificate Creation and Replacement"
- Page "BlueField Modes of Operation Configuration"

- Section "Redfish Event Log"
- Cmds 0xa8 0x00 and 0xa8 0x01 to "NVIDIA OEM Commands"

Updated:

- Section "Changing Default Password"
- Section "BIOS CA Certificates"
- Section "System Commands"
- Section "SEL Redfish Commands"
- Section "Get Maximum Power Capacity"
- Section "Get Dump Task State"
- Section "BMC and CEC Firmware Update"

Rev v23.10 – November 30, 2023

Added:

- Section "Power Capping"
- Section "DPU Chassis"
- Section "BlueField Console Log"
- Section "Viewing Currently Installed CA Certificates"
- Section "CA Certificates Collection Modification"
- Section "Enable RShim on DPU BMC"
- Section "Network Management Redfish Commands"
- Section "CEC Update" under "BMC and CEC Firmware Update"
- Section "OOB Network 3-Port Switch Control"

- Appendix "Provisioning Software Upgrade Flow"

Updated:

- Section "Boot Config Using Redfish"
- Section "Installing BFB"
- Section "Factory Reset BMC"
- Section "Reset or Reboot BMC"
- Section "BMC Sensor Data"
- Section "Retrieving Data from BlueField Via IPMB"
- Section "Serial Over LAN (SOL)"
- Section "BMC Dump Operations"

Rev v23.09 – September 20, 2023

Added:

- Section "System Inventory"
- Section "DPU Chassis"
- Section "NIC Subsystem Management"
- Section "Table of Common Commands"

Updated:

- Section "FRU Reading"
- Section "System Logs"
- Section "List of IPMI Supported FRUs"
- Section "Boot Configuration"

- Section "BIOS Secure Boot Configuration"
- Section "BIOS Configuration"
- Section "Reset Control"

Rev v23.07 – August 10, 2023

Added:

- Section "Changing Default Password"
- Section "Account Service"
- Section "Configuring BIOS Secure Boot"
- Section "Configuring BIOS"
- Section "Redfish Certificate Management"
- The commands 0x32 0x97 and 0x32 0x98 to "NVIDIA Custom Commands"

Updated:

- Note in section "Network Protocol Support"
- Section "Boot Configuration"
- Section "Installing BFB"
- Section "BMC and CEC Firmware Update" and its subsections

Rev v23.04 – May 17, 2023

Added:

- Figure "NVIDIA® BlueField®-3 BMC Connector" to section "BMC Console Interface"
- Section "SEL Messages"
- Section "Updating BMC and Glacier Firmware with Vendor Field Mode" which is relevant for NVIDIA® BlueField®-3 DPU only

- Page "Serial Redirect Mode"
- Section "BlueField BMC Redfish Triggers"
- Command 0x32 0x92 and 0x32 0x93 to "NVIDIA Custom Commands" table

Updated:

- Section "BMC Management Interface" with new password requirements
- Section "Sensor Data Record (SDR) Repository"
- Link status codes to the p0_link and p1_link sensors in section "List of IPMI Supported Sensors"
- Section "BMC and CEC Firmware Update"

Rev 2.8.2-34 – October 21, 2022

Added:

- Page "Vendor Field Mode"
- Section "DPU Reset"

Updated:

- Section "Boot Configuration" with note on DPU boot override setting

Rev 2.8.2 – June 01, 2022

Updated:

- NIC thermal sensors line in table under s ection "SDR Entry List"

Rev 2.8.2 – April 04, 2022

Updated:

- Page "NVIDIA OEM Commands"