



NVIDIA DGX B200 Firmware Update Guide

NVIDIA Corporation

Feb 27, 2025

Contents

| | | |
|----------|--|-----------|
| 1 | About Firmware Updates | 1 |
| 1.1 | Firmware Updatable Components | 1 |
| 1.2 | Firmware Update Prerequisites | 2 |
| 1.3 | Firmware Update Methods | 2 |
| 1.4 | Firmware Update Activation | 3 |
| 2 | About the nvfwupd Command | 5 |
| 2.1 | Requirements | 5 |
| 2.2 | Syntax | 5 |
| 2.3 | Understanding Background Processing | 7 |
| 3 | DGX B200 System Firmware Update Guide Version 25.02.6 | 9 |
| 3.1 | Highlights | 9 |
| 3.1.1 | Added Support | 9 |
| 3.1.2 | BMC Fixes | 9 |
| 3.1.3 | SBIOS Fixes | 9 |
| 3.1.4 | The nvfwupd Command Updates | 9 |
| 3.2 | Firmware Package Details | 10 |
| 3.3 | Firmware Update Procedure | 12 |
| 4 | Firmware Changes for NVIDIA DGX B200 Systems | 13 |
| 4.1 | BMC Changes for DGX B200 Systems | 13 |
| 4.1.1 | Changes in 25.02.12 | 13 |
| 4.2 | SBIOS Changes for DGX B200 Systems | 13 |
| 4.2.1 | Changes in 1.6.7 | 13 |
| 4.2.2 | Changes in 1.6.6 | 13 |
| 4.3 | nvfwupd Command Changes | 13 |
| 4.3.1 | Changes in 2.0.5 | 13 |
| 5 | Firmware Update Steps | 15 |
| 5.1 | Before You Begin | 15 |
| 5.2 | Update Steps | 15 |
| 5.2.1 | Firmware Update Output of mlxfwmanager | 19 |
| 6 | Known Issues | 25 |
| 6.1 | Functional Issues | 25 |
| 6.2 | VBIOS Incompatibility Issue | 25 |
| 6.2.1 | Issue | 25 |
| 6.2.2 | Explanation | 26 |
| 6.2.3 | Workaround | 26 |
| 6.3 | Misleading Messages During Firmware Update | 26 |
| 6.3.1 | Issue | 26 |
| 6.3.2 | Workaround | 27 |
| 6.4 | Firmware Inventory Can Be Invalid During Boot | 27 |

| | | |
|-----------|--|-----------|
| 6.4.1 | Issue | 27 |
| 6.4.2 | Workaround | 27 |
| 6.5 | BMC Slow Startup After AC Power Cycle | 27 |
| 6.5.1 | Issue | 27 |
| 6.5.2 | Workaround | 27 |
| 6.6 | Temperature Sensors Can Report No Reading | 27 |
| 6.6.1 | Issue | 27 |
| 6.6.2 | Workaround | 28 |
| 7 | Getting Started with nvdebug | 29 |
| 7.1 | Requirements | 30 |
| 7.2 | The nvdebug Command-Line Interface | 30 |
| 7.2.1 | Syntax | 31 |
| 7.2.2 | The Configuration Files | 32 |
| 7.3 | HGX B200 8-GPU Example | 32 |
| 7.4 | DGX Platform Example | 33 |
| 7.5 | Redfish Collectors | 35 |
| 7.6 | IPv6 Configuration | 36 |
| 8 | Viewing the Installed Firmware and Package Versions | 37 |
| 9 | Updating the BMC | 41 |
| 10 | Firmware Update of Motherboard Tray: All Components | 43 |
| 11 | Updating BMC EROT on the Motherboard Tray | 45 |
| 12 | Updating SBIOS EROT on the Motherboard Tray | 47 |
| 13 | Updating the BIOS on the Motherboard Tray | 49 |
| 14 | Updating the CPLDs on the Motherboard Tray | 51 |
| 15 | Updating the Midplane CPLDs on the Motherboard Tray | 53 |
| 16 | Updating the NVMe Firmware | 55 |
| 17 | Updating the Power Supply Units on the Motherboard Tray | 57 |
| 18 | Updating the PCIe Switches on the Motherboard Tray | 59 |
| 19 | Updating the PCIe Retimers on the Motherboard Tray | 61 |
| 20 | Updating the ConnectX-7 Firmware | 63 |
| 21 | Updating the Intel NIC Firmware | 65 |
| 22 | Firmware Update of GPU Tray: All Components | 67 |
| 23 | Platform-Agnostic Firmware Updates | 69 |
| 23.1 | Configuration Parameters | 69 |
| 23.2 | Running nvfwupd Using the Configuration File | 70 |
| 24 | Performing a Factory Reset | 75 |
| 25 | Forcing a Firmware Downgrade | 77 |
| 25.1 | Prerequisites | 77 |

| | | |
|-----------|---|-----------|
| 25.2 | Procedure | 77 |
| 26 | Troubleshooting an Unsuccessful Firmware Update | 79 |
| 26.1 | Firmware Update Terminates due to Component Not Found | 79 |
| 26.2 | No Devices Where Detected for Handle ID 0 | 80 |
| 26.3 | Wait for Firmware Update Started ID | 80 |
| 27 | Version 25.02.5 | 81 |
| 27.1 | Highlights | 81 |
| 27.1.1 | Added Support | 81 |
| 27.1.2 | BMC Fixes | 81 |
| 27.1.3 | SBIOS Fixes | 81 |
| 27.1.4 | The nvfwupd Command Updates | 81 |
| 27.2 | Firmware Package Details | 82 |
| 27.3 | Firmware Update Procedure | 84 |
| 28 | Third-Party License Notices | 85 |
| 28.1 | Micron msecli | 85 |
| 28.2 | Mellanox (OFED) | 86 |
| 29 | Notices | 87 |
| 29.1 | Notice | 87 |
| 29.2 | Trademarks | 88 |
| 30 | About Firmware Updates | 89 |
| 30.1 | Firmware Updatable Components | 89 |
| 30.2 | Firmware Update Prerequisites | 90 |
| 30.3 | Firmware Update Methods | 90 |
| 30.4 | Firmware Update Activation | 91 |

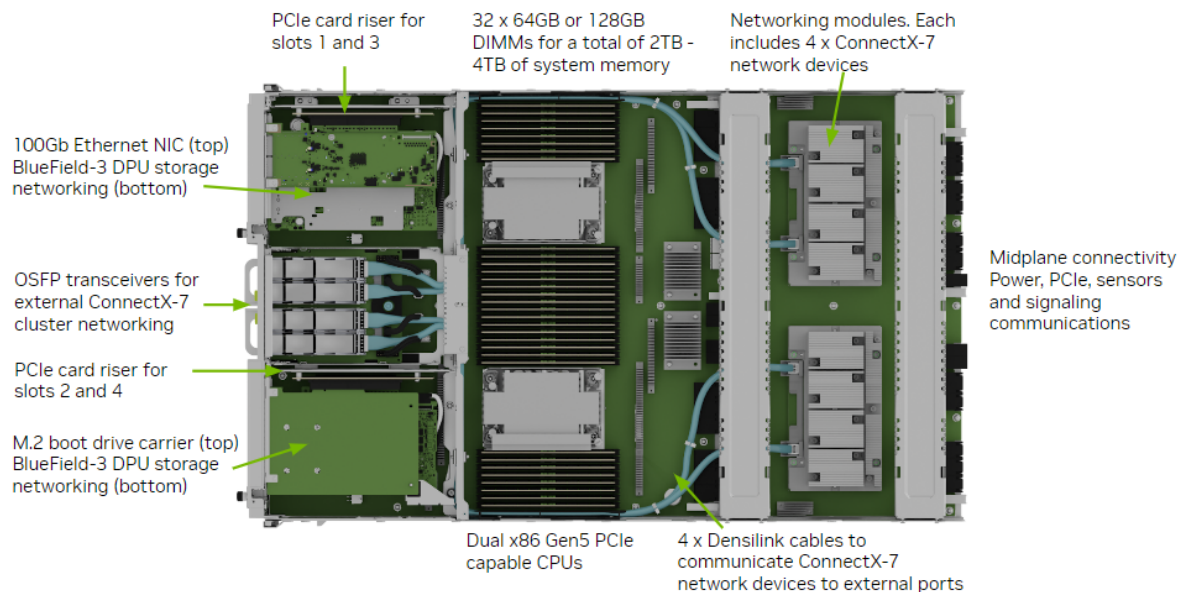
Chapter 1. About Firmware Updates

This topic provides an overview of firmware updates on the NVIDIA DGX™ B200 system.

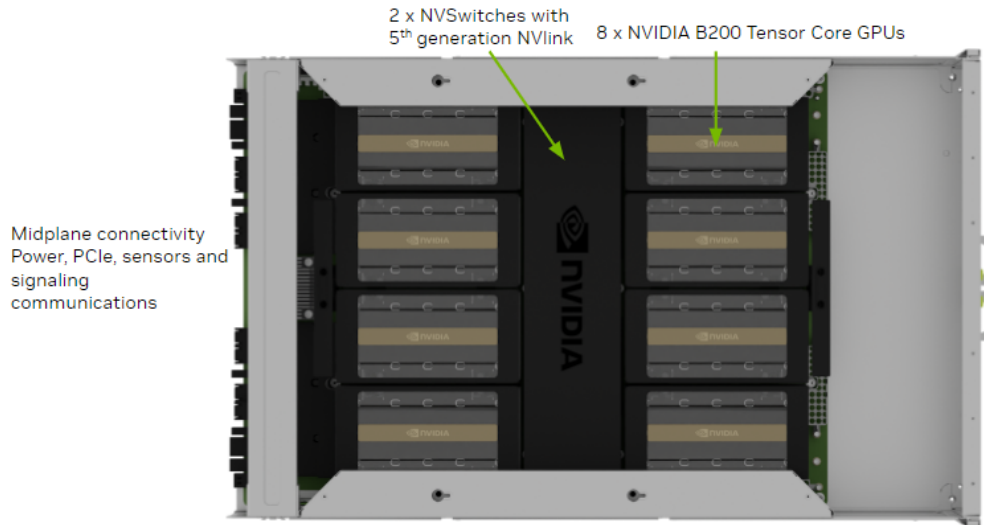
1.1. Firmware Updatable Components

The NVIDIA DGX™ B200 system has several firmware updatable components. Some of the components are on the following two trays in the system:

- ▶ The motherboard tray has components, such as the CPUs, PCH, BMC as shown in the following figure:



- ▶ The GPU tray has components, such as the GPUs, NVSwitches, HMC as shown in the following figure:



You can update the firmware on the NVIDIA DGX B200 system components out-of-band (OOB) by using Redfish APIs or from the host operating system by using the command-line interface (CLI) commands.

1.2. Firmware Update Prerequisites

- ▶ You can download firmware packages from the [NVIDIA Enterprise Support Portal](#).
- ▶ You must know the BMC IP address, a user name, and a password. The sample commands in this document show `admin` for both the user name and the password.
- ▶ You must have the `nvfwupd` executable or know how to use the Redfish API.

1.3. Firmware Update Methods

Most of the sample commands in this document show how to update firmware using the `nvfwupd` command. You can download the executable from the [NVIDIA Enterprise Support Portal](#).

- ▶ For more information about the command, refer to [About the nvfwupd Command](#).
- ▶ For best practice when updating the firmware, follow the instructions in [Firmware Update Steps](#).

You can run the `nvfwupd` command interactively to update systems. Most command examples in this document show this interactive approach. If you have several systems to update, you can create a JSON file that identifies the systems to update. Refer to [Platform-Agnostic Firmware Updates](#) for more information.

An alternative to the `nvfwupd` command is to update firmware by using the Redfish API. The BMC network interface provides remote management with Redfish APIs.

The [Known Issues](#) for updating firmware and the [firmware update steps](#) still apply when you use the Redfish API.

Refer to [Redfish APIs Support](#) in the *NVIDIA DGX B200 System User Guide* for more information and sample commands. The sample commands show how to update firmware with the `curl` command.

1.4. Firmware Update Activation

After the firmware update, you must perform one or more of the following tasks to activate the firmware update, depending on the components being updated:

- ▶ BMC component

Reset the BMC by running the following command:

```
sudo ipmitool mc reset cold
```

- ▶ PCIe Switch, PCIe Retimer, BIOS, and HGX (GPU Tray) components

Perform a cold reset on the system using the following command:

```
sudo ipmitool chassis power cycle
```

- ▶ EROT and CPLD components

Perform an AC power cycle on the system by unplugging all the power supplies and then reconnecting them either manually or through an external PDU device.

Note

The AC power cycle will activate firmware for all updated components.

Chapter 2. About the nvfwupd Command

2.1. Requirements

The nvfwupd executable runs on the Linux operating system and is available for x86_64 or arm64 architecture machines. You can run the x86_64 executable locally on the DGX system or use a remote Linux system.

If you run nvfwupd locally on the DGX system, instead of connecting to the BMC IP address, as shown in the sample commands, you can connect to the host Redfish interface IP address.

To download the latest version of the nvfwupd executable, log in to the [NVIDIA Enterprise Support Portal](#).

2.2. Syntax

```
nvfwupd version 2.0.5

Usage: nvfwupd [ global options ] <command>

Global options:
  -t --target ip=<BMC IP> user=<BMC login id> password=<BMC password> servertime=
  ↪<Type of server>
      BMC target comprising BMC IP address and BMC login credentials. The
  ↪servertime sub-option is optional.
      Valid value for servertime is one of [DGX, HGX, MGX, GH200, NVOS, HGXB100,
  ↪GB200, MGX-NVL, GB200Switch].

  -c --config Path for config file (optional).
      Configure tool behavior.

  -v --verbose Chosen path for logfile (optional). Default path is current working
  ↪directory.
      Increase verbosity.

Commands:
  help          Show tool help.
```

(continues on next page)

(continued from previous page)

```

version      Show tool version.

show_pkg_content [ options... ]
  -p --package      PLDM firmware package.

unpack [ options... ]
  -p --package      PLDM firmware package.
  -o --outdir       Directory path to save unpacked firmware files
↳(optional).
                    Default path is current working directory of tool.

<Global options...> show_version [ options... ]
  -p --package      PLDM firmware package.
  -j --json         Show output in JSON.

<Global options...> update_fw [ options... ]
  -p --package      PLDM firmware package.
  -y --yes          Bypass firmware update confirmation prompt.
  -b --background  Exit without waiting for the update process to
↳finish.
  -t --timeout      API request timeout value in seconds.
  -s --special      Special Update json file.
  -d --details      Show update progress in table format.
  -j --json         Show output in JSON. Must be paired with the -b
↳confirmation prompt.
                    background option, and always bypasses update

<Global options...> activate_fw [ options... ]
  -c --cmd          Activation command name.
↳OFF', 'PWR_ON',
                    List of supported commands ['PWR_STATUS', 'PWR_
↳PWR_CYCLE',
                    'PWR_CYCLE', 'RESET_COLD', 'RESET_WARM', 'NVUE_
↳RF_AUX_PWR_CYCLE'].

<Global options...> force_update [ options... ]
  enable|disable|status Enable, disable, or check current force update
↳value on target.
  -j --json         Show output in JSON.

<Global options...> show_update_progress [ options... ]
  -i --id           List of Task IDs delimited by space.
  -j --json         Show output in JSON.

<Global options...> perform_factory_reset

<Global options...> install_license

<Global options...> make_upd_targets [ options... ]
  -o --outdir       Directory path to create update target files
↳(optional).
                    Default path is current working directory of
↳tool.

```

2.3. Understanding Background Processing

By default, the `nvfwupd update_fw` command communicates with the Redfish API, prints the firmware update progress to the console, and then exits when the command runs to completion.

Alternatively, you can specify the `--background` or `-b` argument so that the command communicates with the Redfish API, prints the task ID to the console, and then exits. In this case, you can periodically run the `nvfwupd show_update_progress` command with the task ID to monitor the update progress.

1. Create a component-specific `updparameters.json` file.
2. Update the firmware in the background:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-user-id> password=<bmc-password> update_
↪fw \
  -p <firmware-package-file> -y -s updparameters.json -b
```

Example output:

```
FW recipe: ['<firmware-package-file>']
{"@odata.type": "#UpdateService.v1_6_0.UpdateService", ...}
FW update started, Task Id: 1
```

Note

If the Platform `dgxhxxx` not supported error appears, choose either of the following methods to resolve the unidentified platform issue:

- ▶ Update the `nvfwupd` tool to version 2.0.4 or later for automatic DGX platform detection.
- ▶ Specify the `servertype=DGX` sub-option with the `--target` option and try again. For example,

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-user-id> password=<bmc-password>
↪\
  servertype=DGX update_fw -p <firmware-package-file> -y -s parameters.json
```

3. Display the update progress for the task:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-user-id> password=<bmc-password> show_
↪update_progress -i 1
```

Example output:

```
{'id': ['1']}
Task Info for Id: 1
StartTime: 2023-03-11T01:10:01-0000
TaskState: Running
PercentComplete: 58
TaskStatus: OK
EndTime: 2023-03-11T01:10:00+00:00
TaskStatus: Task /redfish/v1/UpdateService/upload is running normally.
```

Eventually, the percent complete field reports `100` and the task state field reports `Completed`.

4. To activate the firmware update, refer to [Firmware Update Activation](#) for more information.

Chapter 3. DGX B200 System Firmware Update Guide Version 25.02.6

3.1. Highlights

3.1.1. Added Support

- ▶ Fixed an issue where a boot order setting via Redfish API did not take effect.

3.1.2. BMC Fixes

- ▶ No BMC fixes for the initial release.

3.1.3. SBIOS Fixes

- ▶ Fixed an issue where a boot order setting via Redfish API did not take effect.

3.1.4. The `nvfwupd` Command Updates

- ▶ Added support for parallel firmware updates through the YAML configuration file.
- ▶ Added the `--json` option to the `update_fw`, `show_update_progress`, and `force_update` commands.
- ▶ Added IPv6 support.
- ▶ Deprecated the `targets` sub-option for multi-target input. Use `config.yaml` input instead.

3.2. Firmware Package Details

This firmware release supports the following systems:

- ▶ NVIDIA DGX B200

This firmware release supports the following operating systems:

- ▶ NVIDIA DGX OS 7.0.1 and higher

For more information about the operating systems, refer to the [NVIDIA Base OS](#) documentation.

You can download firmware packages from the [NVIDIA Enterprise Support Portal](#).

The following table shows the firmware package files:

| Components | Sample File Name |
|--|---|
| Combined archive | DGXB200_25.02.6.tar.gz The combined archive includes the firmware for the system components and the firmware for the GPU tray. |
| <ul style="list-style-type: none"> ▶ Motherboard tray package ▶ GPU tray transition package ▶ GPU tray latest package | <ul style="list-style-type: none"> ▶ nvfw_DGX_250220.1.0.fwpkg ▶ nvfw_DGX-HGX-B100-B200x8_250114.1.0.fwpkg ▶ nvfw_DGX-HGX-B100-B200x8_250124.1.3.fwpkg |

The following table shows the information about component firmware versions.

| Component | Version |
|--|--|
| Host BMC | 25.02.12 |
| Host BMC ERoT | 04.0058 |
| SBIOS ERoT | 04.0058 |
| SBIOS | 1.6.7 Refer to SBIOS Changes for DGX B200 Systems for the list of changes. |
| Motherboard CPLD | 0.2.1.9 |
| Midplane CPLD | 0.2.1.3 |
| PSU (Delta ECD16020137) | Primary 0204 Secondary 0201 Community 0204 |
| LiteOn | 0.5.0.5 |
| Broadcom Gen5 PCIe Switch (PEX89072-B01) | Switch 0: 0.0.7 Switch 1: 1.0.7 |
| Astera Labs Gen5 PCIe Retimer (PT5161L) | 2.07.19 |
| Network (Cluster) Card - ConnectX-7 | 28.43.2026 |
| Network (Storage) Card - ConnectX-7 | 28.43.2026 |
| Network Card - BlueField-3 | 32.43.2024 |
| VBIOS | 97.00.7F.00.01 |
| NVSwitch (GPU Tray) | 35.2014.1638 |
| ERoT (GPU Tray) | 02.0192 |
| HMC (GPU Tray) | HGX-24.11-1-rc57 |
| FPGA (GPU Tray) | 1.61 |
| PCIe Switch (GPU Tray) | 1.9.5F |
| Astera Labs Gen5 PCIe Retimer (GPU Tray) (PT5161L) | 2.7.20 |
| Intel 10G Ethernet | v3.60 |
| 3.2. Firmware Package Details | v4.50 |
| Intel Ethernet Network Adapter (E810-C-Q2) | |

3.3. Firmware Update Procedure

Refer to *Firmware Update Steps*.

Chapter 4. Firmware Changes for NVIDIA DGX B200 Systems

4.1. BMC Changes for DGX B200 Systems

4.1.1. Changes in 25.02.12

- ▶ The initial BMC firmware version.

4.2. SBIOS Changes for DGX B200 Systems

4.2.1. Changes in 1.6.7

- ▶ Fixed an issue where a boot order setting via Redfish API did not take effect.

4.2.2. Changes in 1.6.6

- ▶ The initial SBIOS firmware version.

4.3. nvfwupd Command Changes

4.3.1. Changes in 2.0.5

- ▶ Added support for parallel firmware updates through the YAML configuration file.
- ▶ Added the `--json` option to the `update_fw`, `show_update_progress`, and `force_update` commands.
- ▶ Added IPv6 support.
- ▶ Deprecated the `targets` sub-option for multi-target input. Use `config.yaml` input instead.

Chapter 5. Firmware Update Steps

5.1. Before You Begin

- ▶ Stop all unnecessary system activity before you begin the firmware update.
- ▶ Stop all GPU activity, including running the `nvidia-smi` command. GPU activity and running the command can prevent the VBIOS update.
- ▶ Do not add additional loads on the system, such as user jobs, diagnostics, or monitoring services, while an update is in progress. A high workload can disrupt the firmware update process and result in an unusable component.
- ▶ When you begin the firmware update, the update software assists in determining the activity state of the DGX system and provides a warning if it detects that activity levels are above a pre-determined threshold. If you encounter the warning, take action to reduce the workload before proceeding with the firmware update.
- ▶ Fan speeds can increase during the BMC firmware update. This increase in speed is a normal part of the BMC firmware update process.

5.2. Update Steps

1. View the installed versions compared with the newly available firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-username> password=<bmc-password> \  
  show_version -p nvfw_DGX_250220.1.0.fwpkg \  
  nvfw_DGX-HGX-B100-B200x8_250114.1.0.fwpkg
```

2. Update the BMC.

1. Create a file, such as `update_bmc.json`, with the following contents:

```
{  
  "Targets" : ["/redfish/v1/UpdateService/FirmwareInventory/HostBMC_0"]  
}
```

2. Run the following command to update the BMC:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-username> password=<bmc-password>  
↪ update_fw \  
  -p nvfw_DGX_250220.1.0.fwpkg -y -s update_bmc.json
```

3. Reboot the BMC.

- ▶ Use the shell on the system:

```
$ ipmitool mc reset cold

# Wait a couple of minutes and confirm the BMC is back online.
$ ipmitool mc info
```

- ▶ Alternatively, you can use the Web UI through a browser.

4. Update the components on the motherboard tray.

For a one-shot firmware update, the BMC will perform a firmware update on all components in the provided bundle, for example, `nvfw_DGX_xxxxxx.x.x.fwpkg`, which includes the Host BMC (if the `force_update` option is specified), Host BIOS, ERoT, PCIe Retimer, PCIe Switch, PSU, Motherboard CPLD, and Midplane CPLD.

1. Create a file, such as `mb_tray.json`, with empty braces:

```
{}
```

2. Update the firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-username> password=<bmc-password>
↪update_fw \
-p nvfw_DGX_250220.1.0.fwpkg -y -s mb_tray.json
```

5. Update the GPU tray to the **intermediate** firmware version using the transition GPU package.

1. Create a `gpu_tray.json` file with the following contents:

```
{
  "Targets" : ["/redfish/v1/UpdateService/FirmwareInventory/HGX_0"]
}
```

2. Update the firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-username> password=<bmc-password>
↪update_fw \
-p nvfw_DGX-HGX-B100-B200x8_250114.1.0.fwpkg -y -s gpu_tray.json
```

This step performs parallel updates on all the components contained in the GPU tray, such as VBIOS, NVSwitch, ERoTs, and FPGA.

3. Verify that the background copy has been completed successfully by looking for "BackgroundCopyStatus": "Completed" in the following command output:

```
curl -s -k -u <bmc-user>:<bmc-password> -H content-type:application/json \
-X GET https://<bmc-ip-address>/redfish/v1/Chassis/HGX_ERoT_BMC_0 | jq
```

4. Perform a cold reset to restart the system:

```
ipmitool chassis power cycle
```

6. Update the GPU tray to the **latest** firmware version.

1. Repeat the firmware update step for the GPU tray using the latest GPU package:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-username> password=<bmc-password>
↪update_fw \
-p nvfw_DGX-HGX-B100-B200x8_250124.1.3.fwpkg -y -s gpu_tray.json
```

2. Perform a DC power cycle on the system and confirm that it boots completely.

Execute a poweroff on the command line in the OS or from the BMC, followed by powering on from the BMC.

7. Confirm the firmware update is complete by viewing the installed versions again.

After the system is operational again, repeat the following command to confirm all firmware has been updated:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-username> password=<bmc-password> \
show_version -p nvfw_DGX_250220.1.0.fwpkg \
nvfw_DGX-HGX-B100-B200x8_250124.1.3.fwpkg
```

8. Execute background copy commands for the BMC and the system BIOS.

1. BMC:

Background copy Redfish API request:

```
curl -k -u <bmc-user>:<password> --request POST --location 'https://<bmc-ip-
↪address>/redfish/v1/UpdateService/Actions/Oem/NvidiaUpdateService.
↪CommitImage' \
--header 'Content-Type: application/json' \
--data '{
    "Targets": ["/redfish/v1/UpdateService/FirmwareInventory/HostBMC_
↪0"]
}'
```

Example response:

```
{
  "@odata.type": "#UpdateService.v1_11_0.UpdateService",
  "Messages": [
    {
      "@odata.type": "#Message.v1_0_8.Message",
      "Message": "A new task /redfish/v1/TaskService/Tasks/1 was created.",
      "MessageArgs": [
        "/redfish/v1/TaskService/Tasks/1"
      ],
      "MessageId": "Task.1.0.New",
      "Resolution": "None",
      "Severity": "OK"
    },
    {
      "@odata.type": "#Message.v1_0_8.Message",
      "Message": "ActivateFirmware Action is initiated.",
      "MessageId": "UpdateService.1.0.StartActivateFirmware",
      "Resolution": "None",
      "Severity": "OK"
    }
  ]
}
```

Query the update status using the task ID, which is 1, as shown in the output response:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-username> password=<bmc-password>
↪show_update_progress -i 1
```

When the status indicates 100% complete, proceed with the next step.

2. SBIOS:

Background copy Redfish API request:

```
curl -k -u <bmc-user>:<password> --request POST --location 'https://<bmc-ip-
↪address>/redfish/v1/UpdateService/Actions/Oem/NvidiaUpdateService.
↪CommitImage' \
  --header 'Content-Type: application/json' \
  --data '{
    "Targets": ["/redfish/v1/UpdateService/FirmwareInventory/
↪HostBIOS_0"]
  }'
```

Find the task ID from the response, which is usually 2, to query the update status:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-username> password=<bmc-password>
↪show_update_progress -i 2
```

When the status indicates 100% complete, proceed with the next step.

3. Perform an AC power cycle on the system by unplugging all the power supplies and then reconnecting them either manually or through an external PDU device.
9. Update firmware on the network cards and NVMe drives.

Note

During the update, the `mlxfwmanager` command will report the ConnectX-7 device identified as `/dev/mst/mt4129_pciconf0` cannot be updated as shown in the following error message:

```
-E- Failed to query /dev/mst/mt4129_pciconf0 device, error : MFE_ICMD_BAD_PARAM
```

This behavior is expected because this device is not one of the networking cards used to cluster the system, but a bridge device used internally and updated using a separate process.

1. To update the ConnectX®-7 cards and NVIDIA® BlueField®-3 cards, navigate to the network directory and run the `mlxfwmanager` command:

```
cd network
sudo mlxfwmanager -u -D .
```

When prompted to update all 10 ConnectX-7 cards and BlueField-3 cards, type Y to confirm. For a complete output, refer to [Firmware Update Output of mlxfwmanager](#).

2. For firmware update on the Intel E810-C Ethernet Network Adapters, refer to [Updating the Intel NIC Firmware](#).
3. For firmware update on the NVMe drives, refer to [Updating the NVMe Firmware](#).

5.2.1. Firmware Update Output of mlxfwmanager

```

$ sudo mlxfwmanager -u -D .
Querying Mellanox devices firmware ...

Device #1:
-----

Device Type:      ConnectX7
Part Number:      --
Description:
PSID:
PCI Device Name:  /dev/mst/mt4129_pciconf0
Base GUID:        N/A
Base MAC:         N/A
Versions:         Current      Available
FW:              --
Status:           Failed to open device

Device #2:
-----

Device Type:      ConnectX7
Part Number:      MCX750500B-0D00_Ax_Bx
Description:      Nvidia adapter card with four ConnectX-7; each up to 400Gb/s IB
↳(default mode) or 400GbE; PCIe 5.0 x32; PCIe switch; crypto disabled; secure boot
↳enabled
PSID:             MT_0000000891
PCI Device Name:  /dev/mst/mt4129_pciconf1
Base GUID:        b8e924030081db74
Versions:         Current      Available
FW:              28.42.1000     28.43.2026
PXE:             3.7.0500      N/A
UEFI:            14.35.0015     N/A
Status:           Update required

Device #3:
-----

Device Type:      ConnectX7
Part Number:      MCX750500B-0D00_Ax_Bx
Description:      Nvidia adapter card with four ConnectX-7; each up to 400Gb/s IB
↳(default mode) or 400GbE; PCIe 5.0 x32; PCIe switch; crypto disabled; secure boot
↳enabled
PSID:             MT_0000000891
PCI Device Name:  /dev/mst/mt4129_pciconf2
Base GUID:        b8e924030081db7c
Versions:         Current      Available
FW:              28.42.1000     28.43.2026
PXE:             3.7.0500      N/A
UEFI:            14.35.0015     N/A
Status:           Update required

Device #4:
-----

Device Type:      ConnectX7

```

(continues on next page)

(continued from previous page)

```

Part Number:      MCX750500B-0D00_Ax_Bx
Description:      Nvidia adapter card with four ConnectX-7; each up to 400Gb/s IB
↳(default mode) or 400GbE; PCIe 5.0 x32; PCIe switch; crypto disabled; secure boot
↳enabled
PSID:            MT_0000000891
PCI Device Name: /dev/mst/mt4129_pciconf3
Base GUID:       b8e924030081db78
Versions:        Current      Available
  FW:            28.42.1000     28.43.2026
  PXE:           3.7.0500      N/A
  UEFI:          14.35.0015    N/A
Status:          Update required
    
```

Device #5:

```

Device Type:      ConnectX7
Part Number:      MCX750500B-0D00_Ax_Bx
Description:      Nvidia adapter card with four ConnectX-7; each up to 400Gb/s IB
↳(default mode) or 400GbE; PCIe 5.0 x32; PCIe switch; crypto disabled; secure boot
↳enabled
PSID:            MT_0000000891
PCI Device Name: /dev/mst/mt4129_pciconf4
Base GUID:       b8e924030081db70
Versions:        Current      Available
  FW:            28.42.1000     28.43.2026
  PXE:           3.7.0500      N/A
  UEFI:          14.35.0015    N/A
Status:          Update required
    
```

Device #6:

```

Device Type:      ConnectX7
Part Number:      MCX750500B-0D00_Ax_Bx
Description:      Nvidia adapter card with four ConnectX-7; each up to 400Gb/s IB
↳(default mode) or 400GbE; PCIe 5.0 x32; PCIe switch; crypto disabled; secure boot
↳enabled
PSID:            MT_0000000891
PCI Device Name: /dev/mst/mt4129_pciconf5
Base GUID:       b8e924030081d954
Versions:        Current      Available
  FW:            28.42.1000     28.43.2026
  PXE:           3.7.0500      N/A
  UEFI:          14.35.0015    N/A
Status:          Update required
    
```

Device #7:

```

Device Type:      ConnectX7
Part Number:      MCX750500B-0D00_Ax_Bx
Description:      Nvidia adapter card with four ConnectX-7; each up to 400Gb/s IB
↳(default mode) or 400GbE; PCIe 5.0 x32; PCIe switch; crypto disabled; secure boot
↳enabled
PSID:            MT_0000000891
    
```

(continues on next page)

(continued from previous page)

```

PCI Device Name: /dev/mst/mt4129_pciconf6
Base GUID:      b8e924030081d95c
Versions:      Current      Available
  FW:         28.42.1000     28.43.2026
  PXE:         3.7.0500      N/A
  UEFI:        14.35.0015    N/A
Status:        Update required

Device #8:
-----

Device Type:    ConnectX7
Part Number:    MCX750500B-0D00_Ax_Bx
Description:    Nvidia adapter card with four ConnectX-7; each up to 400Gb/s IB
↳(default mode) or 400GbE; PCIe 5.0 x32; PCIe switch; crypto disabled; secure boot
↳enabled
PSID:          MT_0000000891
PCI Device Name: /dev/mst/mt4129_pciconf7
Base GUID:      b8e924030081d958
Versions:      Current      Available
  FW:         28.42.1000     28.43.2026
  PXE:         3.7.0500      N/A
  UEFI:        14.35.0015    N/A
Status:        Update required

Device #9:
-----

Device Type:    ConnectX7
Part Number:    MCX750500B-0D00_Ax_Bx
Description:    Nvidia adapter card with four ConnectX-7; each up to 400Gb/s IB
↳(default mode) or 400GbE; PCIe 5.0 x32; PCIe switch; crypto disabled; secure boot
↳enabled
PSID:          MT_0000000891
PCI Device Name: /dev/mst/mt4129_pciconf8
Base GUID:      b8e924030081d950
Versions:      Current      Available
  FW:         28.42.1000     28.43.2026
  PXE:         3.7.0500      N/A
  UEFI:        14.35.0015    N/A
Status:        Update required

Device #10:
-----

Device Type:    BlueField3
Part Number:    900-9D3B6-00CN-A_Ax
Description:    NVIDIA BlueField-3 B3240 P-Series Dual-slot FHHL DPU; 400GbE / NDR
↳IB (default mode); Dual-port QSFP112; PCIe Gen5.0 x16 with x16 PCIe extension
↳option; 16 Arm cores; 32GB on-board DDR; integrated BMC; Crypto Enabled
PSID:          MT_0000000883
PCI Device Name: /dev/mst/mt41692_pciconf0
Base GUID:      b8e9240300a65e18
Base MAC:       b8e924a65e18
Versions:      Current      Available
  FW:         32.42.1000     32.43.2024

```

(continues on next page)

(continued from previous page)

```

PXE:          3.7.0500      N/A
UEFI:         14.35.0015   N/A
UEFI Virtio blk: 22.4.0013   N/A
UEFI Virtio net: 21.4.0013   N/A
Status:       Update required

Device #11:
-----

Device Type:   BlueField3
Part Number:   900-9D3B6-00CN-A_Ax
Description:   NVIDIA BlueField-3 B3240 P-Series Dual-slot FHHL DPU; 400GbE / NDR
↳IB (default mode); Dual-port QSFP112; PCIe Gen5.0 x16 with x16 PCIe extension
↳option; 16 Arm cores; 32GB on-board DDR; integrated BMC; Crypto Enabled
PSID:         MT_0000000883
PCI Device Name: /dev/mst/mt41692_pciconf1
Base GUID:    b8e9240300a667be
Base MAC:     b8e924a667be
Versions:     Current      Available
FW:          32.42.1000     32.43.2024
PXE:         3.7.0500      N/A
UEFI:        14.35.0015    N/A
UEFI Virtio blk: 22.4.0013   N/A
UEFI Virtio net: 21.4.0013   N/A
Status:       Update required

-----
-E- Failed to query /dev/mst/mt4129_pciconf0 device, error : MFE_ICMD_BAD_PARAM
Found 10 device(s) requiring firmware update...

Perform FW update? [y/N]: y
Device #1: Device query failed
Device #2: Updating FW ...
FSMST_INITIALIZE - OK
Writing Boot image component - OK
Done
Device #3: Updating FW ...
FSMST_INITIALIZE - OK
Writing Boot image component - OK
Done
Device #4: Updating FW ...
FSMST_INITIALIZE - OK
Writing Boot image component - OK
Done
Device #5: Updating FW ...
FSMST_INITIALIZE - OK
Writing Boot image component - OK
Done
Device #6: Updating FW ...
FSMST_INITIALIZE - OK
Writing Boot image component - OK
Done
Device #7: Updating FW ...
FSMST_INITIALIZE - OK
Writing Boot image component - OK

```

(continues on next page)

(continued from previous page)

```
Done
Device #8: Updating FW ...
FSMST_INITIALIZE - OK
Writing Boot image component - OK
Done
Device #9: Updating FW ...
FSMST_INITIALIZE - OK
Writing Boot image component - OK
Done
Device #10: Updating FW ...
FSMST_INITIALIZE - OK
Writing Boot image component - OK
Done
Device #11: Updating FW ...
FSMST_INITIALIZE - OK
Writing Boot image component - OK
Done
```

```
Restart needed for updates to take effect.
-E- One or more errors were encountered.
```

Chapter 6. Known Issues

- ▶ *Functional Issues*
- ▶ *VBIOS Incompatibility Issue*
- ▶ *Misleading Messages During Firmware Update*
- ▶ *Firmware Inventory Can Be Invalid During Boot*
- ▶ *BMC Slow Startup After AC Power Cycle*
- ▶ *Temperature Sensors Can Report No Reading*

6.1. Functional Issues

- ▶ You cannot update firmware of the individual components of the DGX B200 GPU tray. For example, you can not individually update the firmware for the GPU only. You must update the firmware by flashing the entire DGX B200 GPU tray.
- ▶ Firmware download is not automatic. You must download the firmware manually from the [NVIDIA Enterprise Support Portal](#).

6.2. VBIOS Incompatibility Issue

6.2.1. Issue

Updating directly from a version earlier than 97.00.5E.00.XX to a version later than 97.00.7C.00.XX might fail. When using the Redfish method, you might see an error similar to the following:

```
{
  "@odata.type": "#MessageRegistry.v1_4_1.MessageRegistry",
  "Message": "Verification of image '97.00.7C.00.05' at HGX_FW_GPU_SXM_4' failed.",
  "MessageArgs": [
    "97.00.7C.00.05",
    "HGX_FW_GPU_SXM_4"
  ],
}
```

(continues on next page)

(continued from previous page)

```
"MessageID": "Update.1.0.VerificationFailed",  
"Resolution": "None."  
"Severity": "Critical"  
}
```

6.2.2. Explanation

VBIOS firmware data structures in versions earlier than 97.00.5E.00.XX and versions later than 97.00.7C.00.XX are incompatible.

6.2.3. Workaround

If your current VBIOS version is:

- ▶ 97.00.5E.00.XX or later:

Update to the latest VBIOS version directly.

- ▶ Earlier than 97.00.5E.00.XX:

Follow these steps:

1. Update to a version between 97.00.5E.00.XX and 97.00.7C.00.XX.
2. Then, update to the latest version.

6.3. Misleading Messages During Firmware Update

6.3.1. Issue

During the process of the ConnectX-7 firmware update, upon completion of applying the update, a reboot is required as suggested by these messages: `To load new FW, run mlxfwreset or reboot machine.` and `Please reboot machine to load new configurations.` However, rebooting the system does not load the firmware update or new configurations properly for the ConnectX-7 firmware versions 28.36.1010 and later.

6.3.2. Workaround

For the firmware update and new configurations to load successfully, perform an AC power cycle on the system instead of rebooting.

6.4. Firmware Inventory Can Be Invalid During Boot

6.4.1. Issue

In rare instances, polling the firmware inventory endpoint of the BMC Redfish API can report an inaccurate firmware versions for the HGX_0 component.

6.4.2. Workaround

Query the firmware inventory after the system completes the boot sequence to retrieve the current firmware inventory.

6.5. BMC Slow Startup After AC Power Cycle

6.5.1. Issue

After an AC power cycle, the BMC can require approximately 10 minutes before it is available for communication. The BMC is typically available within three minutes.

6.5.2. Workaround

No workaround is available.

6.6. Temperature Sensors Can Report No Reading

6.6.1. Issue

The following sensors can report No Reading rather than a temperature value:

- ▶ TEMP_PSU4
- ▶ TEMP_PSU5
- ▶ PWR_PSU5
- ▶ SPD_FAN_PSU5_R
- ▶ SPD_FAN_PSU5_R
- ▶ STATUS_PSU0
- ▶ STATUS_PSU1
- ▶ STATUS_PSU2
- ▶ STATUS_PSU3
- ▶ STATUS_PSU4
- ▶ STATUS_PSU5
- ▶ STATUS_HMC
- ▶ TEMP_PCIE_SW_1
- ▶ TEMP_Cedar_OSFP0
- ▶ TEMP_Cedar_OSFP1
- ▶ TEMP_Cedar_OSFP2
- ▶ TEMP_Cedar_OSFP3
- ▶ TEMP_PCIE_CX7_1
- ▶ TEMP_PCIE_CX7_2
- ▶ TEMP_CX7_QSFP0
- ▶ TEMP_CX7_QSFP1
- ▶ TEMP_CX7_QSFP2
- ▶ TEMP_CX7_QSFP3
- ▶ TEMP_Intel_NIC
- ▶ TEMP_NIC_QSFP0
- ▶ TEMP_NIC_QSFP1

6.6.2. Workaround

Polling the sensors again can resolve the issue.

Chapter 7. Getting Started with nvdebug

The NVIDIA® NVDebug tool, `nvdebug`, runs on server platforms or from remote client machines. This binary tool, which is available for `x86_64` or `arm64-SBSA` architecture systems, collects the following information:

- ▶ Out-of-band (OOB) BMC logs and information for troubleshooting server issues
- ▶ Logs from the host

7.1. Requirements

Table 1: Requirements for Client Host and Server Host

| Requirement | Client Host | Server Host |
|--|-------------|-------------|
| Linux-based operating system: Linux kernel 4.4 or later supported (version 4.15 or later recommended) | X | X |
| GNU C Library glibc-2.7 or later | X | X |
| OS: Ubuntu 18.04 or later supported (Ubuntu 22.04 recommended) | X | X |
| Python 3.10 | X | X |
| ipmitool 1.8.18 or later | X | X |
| The sshpass command | X | X |
| A server device under test (DUT) accessible by the BMC from the client host using Redfish and IPMI-over-LAN. | X | X |
| The <code>nvme-cli</code> tool | | X |
| BMC Management and Server Host Management networks are in the same subnet. | | X |

NVSwitch tray host requires NVOS version 2.

7.2. The nvdebug Command-Line Interface

The high-level syntax of the `nvdebug` command supports the collection of debug logs over OOB.

You can run the tool in either of the following ways:

- ▶ From a remote machine with access to the BMC and host.
- ▶ Directly on the host machine if the host can access the BMC.

If the host IP is passed through the configuration file or the command-line interface (CLI) using `-I/--hostip`, `nvdebug` assumes the tool runs on a remote machine. Otherwise, `nvdebug` assumes the tool runs on the host and collects the host logs locally.

7.2.1. Syntax

```
$ nvdebug -i <BMCIP> -u <BMCUSER> -p <BMCPASS> -t <PLATFORM>
```

Mandatory options:

- i/--ip is the BMC IP address.
- u/--user is BMC username with administrative privileges.
- p/--password is BMC administrative user password.
- t/--platform is the platform type of the DUT, and it accepts DGX, HGX-HMC, arm64, x86_64, and NVSwitch.

Additional credentials:

- r/--sshuser is BMC SSH username.
- w/--sshpass is BMC SSH password.
- R/--rfuser is BMC Redfish username.
- W/--rfpass is BMC Redfish password.

Host options:

- I/--hostip is the Host IP Address.
If the IP address is not provided, the tool assumes it is running on the host machine.
- U/--hostuser is the Host username with administrative privileges.
- H/--hostpass is the Host password.

Additional options:

- b/--baseboard <baseboard> is the baseboard type, such as Hopper-HGX-8-GPU and Blackwell-HGX-8-GPU.
- C/--config <file path> is the path to the config file. The default is ./config.yaml.
- d/--dutconfig <dut config path> is the path to the DUT specific config file. The default path is ./dut_config.yaml.
- c/--common collects the common logs using the included common.json file.
- v/--verbose displays the detailed output and error messages.
- o/--outdir <output dir> the output directory where the output is generated. The default location is /tmp.
- P/--port <fw_port> is the port number that will be used for forwarding. The --port variable applies only to HGX-Baseboard based platforms, and the default value is 18888.
- local enables Local Execution mode.
- z/--skipzip skips zipping individual DUT folders.

Log collection options:

- S/--cids CID [CID ...] runs the log collectors that correspond to the CIDs that were passed.
- g/--loggroup <Redfish|IPMI|SSH|Host|HealthCheck> runs all log collectors of a specific type that is supported on the current platform. Only one collector group can be specified.
- j/--vendor_file <vendor.json> is a vendor-defined JSON file that uses proprietary methods and tools as defined by the user. The -S and -g options cannot be used together.

Utility options:

- h/--help and --version are standalone options, and -l/--list requires the platform

(continues on next page)

(continued from previous page)

```

type to be specified using -t/--platform.
--parse <log dump> parses an nvdebug log dump and decodes the binary data.
-h/--help provides information about tool usage.
--version displays the current version of the tool.
-l/--list [Redfish|IPMI|SSH|Host|HealthCheck] lists log collectors that are
↳ supported by platform
   with their collector IDs (CID). If a type is passed, it will only list log
↳ collectors
   of that type. The -l/--list options require the target platform type to be
↳ specified with -t/--platform.

```

```

By default, if option -c is not included, the nvdebug tool will collect logs
↳ based on the common.json
   and platform_xyz.json files. At the end of the run, the tool will generate the
↳ output log xyz.zip
   file in the directory specified by the -o option. If no directory is provided,
↳ the log
   will be generated in the /tmp directory.

```

7.2.2. The Configuration Files

The NVDebug tool has two configuration files in the same folder as the executable:

- ▶ The DUT configuration file: The default is `dut_config.yaml`.
- ▶ The NVDebug-specific configuration file: The default is `config.yaml`.

These files can be used to provide additional (but optional) configuration data. If an argument is provided by both the CLI and the configuration file, the value provided through the CLI takes precedence.

7.3. HGX B200 8-GPU Example

To communicate with the HGX baseboard, you need the BMC SSH credentials to set up SSH tunneling through the BMC. By default, the SSH credentials are assumed to be the same as the BMC credentials. To use different credentials, specify the `-r` and `-w` CLI options for the SSH username and password, respectively.

```

nvdebug -i $BMCIP -u $BMCUSER -p $BMCPASS -r SSHUSER -w SSHPASS -t HGX-HMC -P port_num

Log directory created at /tmp/nvdebug_logs_30_09_2024_12_27_46
Starting a collection for DUT dut-1
hgx-b200-node2: [12:28:13] Identified system as Model: P2312-A04, Partno: 692-22312-
↳ 0001-000, Serialno:1324623011823
hgx-b200-node2: [12:28:13] User provided platform type: HGX-HMC
hgx-b200-node2: [12:28:13] BMC IP: XXXX

Log collection has started for dut-1
hgx-b200-node2: [12:45:43] Log collection is now complete
hgx-b200-node2: [12:45:43] Log collection took 17m 30.29s
DUT hgx-b200-node2 completed.

```

(continues on next page)

(continued from previous page)

The log zip file (:literal:`nvdebug_logs_30_09_2024_12_27_46.zip`) will be created **in** the :literal:`/tmp` directory.

The SSH tunnel is set up automatically by the tool using the specified port, and the default value is 18888. To use an existing SSH tunnel, do not set up SSH tunnelling in the configuration file, as shown in the following dut_config file:

```
hgx-b200-node2:
  <<: *dut_defaults
  BMC_IP: "bmc_ip"
  BMC_USERNAME: "bmc_user"
  BMC_PASSWORD: "bmc_pass"
  BMC_SSH_USERNAME: "ssh_user"
  BMC_SSH_PASSWORD: "ssh_pass"
  TUNNEL_TCP_PORT: "port_num"

  SETUP_PORT_FORWARDING: false
```

After configuring the NVDebug tool, run the nvdebug command:

Note

The Host BMC needs to support port forwarding.

Example output:

```
$ nvdebug

Log directory created at /tmp/nvdebug_logs_30_09_2024_12_27_46
Starting a collection for DUT hgx-b200-node2
hgx-b200-node2: [12:28:13] Identified system as Model: P2312-A04, Partno: 692-22312-
↳0001-000, Serialno:1324623011823
hgx-b200-node2: [12:28:13] User provided platform type: HGX-HMC
hgx-b200-node2: [12:28:13] BMC IP: XXXX

Log collection has started for hgx-b200-node2
hgx-b200-node2: [12:45:43] Log collection is now complete
hgx-b200-node2: [12:45:43] Log collection took 17m 30.29s
DUT hgx-b200-node2 completed.

The log zip file (nvdebug_logs_30_09_2024_12_27_46.zip) will be created in the /tmp
↳directory.
```

7.4. DGX Platform Example

To list the collectors that are available on a DGX platform, specify the `-l` option and the `-t DGX` option for log collectors and the DGX platform, respectively:

```
$ nvdebug -l -t DGX
```

Example output:

| Redfish | | |
|---------|---------------------------------|---------------------------------------|
| CID | Collector Name | Log Location |
| R8 | firmware_inventory | Redfish_R8_firmware_inventory.json |
| R9 | firmware_inventory_expand_query | Redfish_R9_firmware_inventory_expand_ |
| ↪ | query.json | |
| R10 | chassis_info | Redfish_R10_chassis_info.json |
| R11 | chassis_expand_query | Redfish_R11_chassis_expand_query.json |
| R12 | system_info | Redfish_R12_system_info.json |
| R13 | system_expand_query | Redfish_R13_system_expand_query.json |
| R14 | manager_info | Redfish_R14_manager_info.json |
| R15 | manager_expand_query | Redfish_R15_manager_expand_query.json |
| R17 | dgx_manager_oem_log_dump | Redfish_R17_dgx_oem_dump_{manager_id} |
| ↪ | _{task_id}.tar.xz | |
| R18 | telemetry_metric_reports | Redfish_R18_report_{metric_report}. |
| ↪ | json | |
| R19 | chassis_thermal_metrics | Redfish_R19_chassis_{chassis}_ |
| ↪ | thermal_metrics.json | |
| R20 | firmware_inventory_table | Redfish_R20_firmware_inventory_table. |
| ↪ | txt | |
| R22 | task_details | Redfish_R22_task_{task_id}.json |
| R23 | nvlink_oob_logs | Redfish_R23_NVLINK_00B_Log_{id}.json |
| R25 | additional_oob_logs | Redfish_R25_00B_Log_{id}.json |
| R26 | chassis_certificates | Redfish_R26_chassis_{chassis_id}_ |
| ↪ | certificate.json | |
| R29 | background_copy_status | Redfish_R29_{chassis_id}_copy_status. |
| ↪ | json | |
| R30 | software_inventory | Redfish_R30_software_inventory |
| R32 | system_post_codes | Redfish_R32_system_post_codes |
| IPMI | | |
| CID | Collector Name | Log Location |
| I1 | mc_info | IPMI_I1_mc_info.txt |
| I2 | lan_info | IPMI_I2_lan_info.txt |
| I3 | session_info | IPMI_I3_session_info.txt |
| I4 | fru_info | IPMI_I4_fru_info.txt |
| I5 | sdr_info | IPMI_I5_sdr_info.txt |
| I6 | sel_info | IPMI_I6_sel_info.txt |
| I7 | sensor_list | IPMI_I7_sensor_list.txt |
| I8 | sel_list | IPMI_I8_sel_list.txt |
| I9 | sel_raw_dump | IPMI_I9_sel_raw_dump.txt |
| I10 | chassis_status | IPMI_I10_chassis_status.txt |
| I11 | chassis_restart_cause | IPMI_I11_chassis_restart_cause.txt |
| I12 | user_list | IPMI_I12_user_list.txt |
| I13 | channel_info | IPMI_I13_channel_info.txt |
| I14 | sdr_elist | IPMI_I14_sdr_elist.txt |
| SSH | | |
| CID | Collector Name | Log Location |
| S2 | bmc_dmesg | BMC_SSH_S2_bmc_dmesg.txt |
| S3 | network_info | BMC_SSH_S3_network_info/... |
| S5 | bmc_list_kernel_modules | BMC_SSH_S5_bmc_list_kernel_modules. |
| ↪ | txt | |
| S8 | bmc_mem_cpu_utilization | BMC_SSH_S8_bmc_mem_cpu_utilization/.. |
| ↪ | . | |
| S11 | uptime | BMC_SSH_S11_uptime.txt |
| S12 | fpga_register_table | BMC_SSH_S12_fpga_register_table.txt |
| S13 | hmc_boot_status | BMC_SSH_S13_hmc_boot_status.txt |

(continues on next page)

(continued from previous page)

| | | |
|-------------|--------------------------|---------------------------------------|
| S15 | bmc_power_status | BMC_SSH_S15_bmc_power_status/... |
| Host | | |
| CID | Collector Name | Log Location |
| H1 | node_dmesg | Host_H1_node_dmesg.tar.gz |
| H2 | node_lspci | Host_H2_node_lspci*.txt |
| H3 | node_smbios | Host_H3_dmidecode*.txt |
| H4 | node_lshw | Host_H4_lshw*.txt |
| H5 | node_nvidia_smi | Host_H5_nvidia-smi*.txt |
| H6 | node_kern_log | Host_H6_node_kern_log.tar.gz |
| H7 | node_crash_dump | Host_H7_node_crash_dump.tar.gz |
| H8 | node_nvme_list | Host_H8_nvme_list-v.txt |
| H9 | node_fabric_manager_log | Host_H9_fabricmanager.log |
| H10 | node_nvflash_log | Host_H10_nvflash_--check_-i_{num}.txt |
| H11 | nvidia_bug_report | Host_H11_nvidia_bug_report_op.log.gz |
| H15 | node_subnet_manager | Host_H15_node_subnet_manager/ |
| H16 | one_diag_dump | Host_H16_one_diag_dump/ |
| H17 | node_nvme_log_dump | Host_H17_nvme_log_dump/ |
| HealthCheck | | |
| CID | Collector Name | Log Location |
| C1 | out_of_band_health_check | HealthCheck_C1_out_of_band_health_ |
| | ↪check.json | |

7.5. Redfish Collectors

To collect only specific collectors, specify the `-S` option for firmware inventory, system information, and ipmi manager information.

```
nvdebug -i <bmc_ip> -u <bmc_user> -p <bmc_pass> ... -t DGX -v -S R8 I1 R12
```

Example output:

```
Log directory created at /tmp/nvdebug_logs_06_11_2024_15_40_27
Starting a collection for DUT dut-1
dut-1: [15:40:34] All preflight checks passed
dut-1: [15:40:34] Identified system as Model: DGXB200, Partno: 965-24387-0002-003,
↪Serialno:1660224000069
dut-1: [15:40:34] User provided platform type: DGX
dut-1: [15:40:34] BMC IP: XXXX
Log collection has started for dut-1
dut-1: [15:40:34]
dut-1: [15:40:34] #####
dut-1: [15:40:34]
dut-1: [15:40:34] Collecting custom logs:
dut-1: [15:40:34]
dut-1: [15:40:34] Log collection was initiated for: r8_firmware_inventory
dut-1: [15:40:36] Log collection for r8_firmware_inventory took 0m 1.71s
dut-1: [15:40:36] Log collection was initiated for: r12_system_info
dut-1: [15:40:36] Log collection for r12_system_info took 0m 0.06s
dut-1: [15:40:36] Log collection was initiated for: i1_mc_info
dut-1: [15:40:36] Log collection for i1_mc_info took 0m 0.14s
dut-1: [15:40:36] Log collection is now complete
```

(continues on next page)

(continued from previous page)

```
dut-1: [15:40:36] Log collection took 0m 2.16s
DUT dut-1 completed.
Log zip created at /tmp/nvdebug_logs_06_11_2024_15_40_27.zip
```

To run the Redfish log collectors, specify the `-g` option for the Redfish log group:

```
$ nvdebug -i $BMC_IP -u $BMC_USER -p $BMC_PASS -t DGX -g Redfish
```

7.6. IPv6 Configuration

By default, the `nvdebug` tool uses IPv4. For IPv6, set `IP_NETWORK` to `ipv6` in the DUT configuration. When providing IPv6 addresses for the BMC/Host, do not use square brackets.

Chapter 8. Viewing the Installed Firmware and Package Versions

Perform the following steps to view the firmware versions that are installed on the system and the versions in the firmware update packages.

- ▶ Compare the installed firmware versions with the versions available in the packages:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-username> password=<bmc-password> \  
  show_version -p nvfw_DGX_250220.1.0.fwpkg \  
  nvfw_DGX-HGX-B100-B200x8_250124.1.3.fwpkg
```

Replace the firmware package file names with the file names that you downloaded.

Tip

If you run the `nvfwupd show_version` command without any arguments, the command displays the currently installed firmware versions.

Example Output

```
System Model: DGXB200  
Part number: xxx-xxxxx-xxxx-xxx  
Serial number: xxxxxxxxxxxxxxx  
BMC IP: 192.168.1.1  
  
Firmware Devices:  
AP Name          Up-To-Date          Sys Version          Pkg Version  
↔ -----          ↔ -----          ↔ -----          ↔ -----  
↔ CPLDMB_0        Yes                  0.2.1.6              0.2.1.6  
↔ CPLDMID_0       Yes                  0.2.0.7              0.2.0.7  
↔ EROT_BIOS_0     No                   00.04.0011.0000_n00  00.04.0018.0000_n00  
↔ EROT_BMC_0     No                   00.04.0011.0000_n00  00.04.0018.0000_n00  
↔ HGX_FW_BMC_0   No                   HGX-22.10-1-rc31     HGX-23.03-09-rc01
```

(continues on next page)

(continued from previous page)

| | | |
|--------------------------|---------------------|---------------------|
| ↪ No | | |
| HGX_FW_ERoT_BMC_0 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_FPGA_0 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_GPU_SXM_1 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_GPU_SXM_2 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_GPU_SXM_3 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_GPU_SXM_4 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_GPU_SXM_5 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_GPU_SXM_6 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_GPU_SXM_7 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_GPU_SXM_8 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_NVSwitch_0 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_NVSwitch_1 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_NVSwitch_2 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_NVSwitch_3 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_ERoT_PCIESwitch_0 | 00.02.0120.0000_n00 | 00.02.0114.0001_n00 |
| ↪ Yes | | |
| HGX_FW_FPGA_0 | 2.0D | 2.09 |
| ↪ Yes | | |
| HGX_FW_GPU_SXM_1 | 96.00.61.00.01 | 96.00.5E.00.00 |
| ↪ Yes | | |
| HGX_FW_GPU_SXM_2 | 96.00.61.00.01 | 96.00.5E.00.00 |
| ↪ Yes | | |
| HGX_FW_GPU_SXM_3 | 96.00.61.00.01 | 96.00.5E.00.00 |
| ↪ Yes | | |
| HGX_FW_GPU_SXM_4 | 96.00.61.00.01 | 96.00.5E.00.00 |
| ↪ Yes | | |
| HGX_FW_GPU_SXM_5 | 96.00.61.00.01 | 96.00.5E.00.00 |
| ↪ Yes | | |
| HGX_FW_GPU_SXM_6 | 96.00.61.00.01 | 96.00.5E.00.00 |
| ↪ Yes | | |
| HGX_FW_GPU_SXM_7 | 96.00.61.00.01 | 96.00.5E.00.00 |
| ↪ Yes | | |
| HGX_FW_GPU_SXM_8 | 96.00.61.00.01 | 96.00.5E.00.00 |
| ↪ Yes | | |
| HGX_FW_NVSwitch_0 | 96.10.35.00.02 | 96.10.2E.00.00 |
| ↪ Yes | | |
| HGX_FW_NVSwitch_1 | 96.10.35.00.02 | 96.10.2E.00.00 |
| ↪ Yes | | |
| HGX_FW_NVSwitch_2 | 96.10.35.00.02 | 96.10.2E.00.00 |
| ↪ Yes | | |
| HGX_FW_NVSwitch_3 | 96.10.35.00.02 | 96.10.2E.00.00 |

(continues on next page)

(continued from previous page)

| | | | |
|---|------------------------|-----------------|----------------|
| ↪ | Yes | | |
| | HGX_FW_PCIERetimer_0 | 1.31.7 | 1.31.7 |
| ↪ | Yes | | |
| | HGX_FW_PCIERetimer_1 | 1.31.7 | 1.31.7 |
| ↪ | Yes | | |
| | HGX_FW_PCIERetimer_2 | 1.31.7 | 1.31.7 |
| ↪ | Yes | | |
| | HGX_FW_PCIERetimer_3 | 1.31.7 | 1.31.7 |
| ↪ | Yes | | |
| | HGX_FW_PCIERetimer_4 | 1.31.7 | 1.31.7 |
| ↪ | Yes | | |
| | HGX_FW_PCIERetimer_5 | 1.31.7 | 1.31.7 |
| ↪ | Yes | | |
| | HGX_FW_PCIERetimer_6 | 1.31.7 | 1.31.7 |
| ↪ | Yes | | |
| | HGX_FW_PCIERetimer_7 | 1.31.7 | 1.31.7 |
| ↪ | Yes | | |
| | HGX_FW_PCIESwitch_0 | 1.7.5F | 1.7.5F |
| ↪ | Yes | | |
| | HGX_InfoROM_GPU_SXM_1 | G520.0200.00.01 | N/A |
| ↪ | No | | |
| | HGX_InfoROM_GPU_SXM_2 | G520.0200.00.01 | N/A |
| ↪ | No | | |
| | HGX_InfoROM_GPU_SXM_3 | G520.0200.00.01 | N/A |
| ↪ | No | | |
| | HGX_InfoROM_GPU_SXM_4 | G520.0200.00.01 | N/A |
| ↪ | No | | |
| | HGX_InfoROM_GPU_SXM_5 | G520.0200.00.01 | N/A |
| ↪ | No | | |
| | HGX_InfoROM_GPU_SXM_6 | G520.0200.00.01 | N/A |
| ↪ | No | | |
| | HGX_InfoROM_GPU_SXM_7 | G520.0200.00.01 | N/A |
| ↪ | No | | |
| | HGX_InfoROM_GPU_SXM_8 | G520.0200.00.01 | N/A |
| ↪ | No | | |
| | HGX_InfoROM_NVSwitch_0 | 5612.0002.00.01 | N/A |
| ↪ | No | | |
| | HGX_InfoROM_NVSwitch_1 | 5612.0002.00.01 | N/A |
| ↪ | No | | |
| | HGX_InfoROM_NVSwitch_2 | 5612.0002.00.01 | N/A |
| ↪ | No | | |
| | HGX_InfoROM_NVSwitch_3 | 5612.0002.00.01 | N/A |
| ↪ | No | | |
| | HostBIOS_0 | 01.00.04 | 01.00.04 |
| ↪ | Yes | | |
| | HostBMC_0 | 23.04.18 | 44.04.19 |
| ↪ | No | | |
| | PCIERetimer_0 | 1.30.12 | 1.30.0 |
| ↪ | Yes | | |
| | PCIERetimer_1 | 1.30.12 | 1.30.0 |
| ↪ | Yes | | |
| | PCIESwitch_0 | 0.0.6 | 00.06.78 |
| ↪ | No | | |
| | PCIESwitch_1 | 1.0.6 | 01.06.78 |
| ↪ | No | | |
| | PSU_0 | 0202.0201.0202 | 0202.0201.0202 |

(continues on next page)

(continued from previous page)

| | | | |
|-------|-----|----------------|----------------|
| ↔ | Yes | | |
| PSU_1 | | 0202.0201.0202 | 0202.0201.0202 |
| ↔ | Yes | | |
| PSU_2 | | 0202.0201.0202 | 0202.0201.0202 |
| ↔ | Yes | | |
| PSU_3 | | 0202.0201.0202 | 0202.0201.0202 |
| ↔ | Yes | | |
| PSU_4 | | 0202.0201.0202 | 0202.0201.0202 |
| ↔ | Yes | | |
| PSU_5 | | 0202.0201.0202 | 0202.0201.0202 |
| ↔ | Yes | | |

HGX Firmware from the GPU tray reports the HGX_InfoRom_GPU_SXM_n and HGX_InfoRom_NVSwitch_n in the firmware inventory Redfish output. As a result, it is included in the preceding output as N/A. In the Up-To-Date column, these entries show No because you cannot update them OOB from the GPU or NVSwitch firmware images respectively.

Chapter 9. Updating the BMC

1. Create a `update_bmc.json` file with the following contents:

```
{
  "Targets" : ["/redfish/v1/UpdateService/FirmwareInventory/HostBMC_0"]
}
```

2. Update the firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-username> password=<bmc-password> update_
↪fw \
  -p nvfw_DGX_250220.1.0.fwpkg -y -s update_bmc.json
```

Example Output

```
FW recipe: ['nvfw_DGX_250220.1.0.fwpkg']
{"@odata.type": "#UpdateService.v1_6_0.UpdateService", "Messages": [{"@odata.type
↪": "#Message.v1_0_8.Message", "Message": "A new task /redfish/v1/TaskService/
↪Tasks/2 was created.", "MessageArgs": ["/redfish/v1/TaskService/Tasks/2"],
↪"MessageId": "Task.1.0.New", "Resolution": "None", "Severity": "OK"}, {"@odata.
↪type": "#Message.v1_0_8.Message", "Message": "The action UpdateService.
↪MultipartPush was submitted to do firmware update.", "MessageArgs": [
↪"UpdateService.MultipartPush"], "MessageId": "UpdateService.1.0.
↪StartFirmwareUpdate", "Resolution": "None", "Severity": "OK"}]}
FW update started, Task Id: 2
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
PercentageComplete: 6
TaskState: Running
PercentComplete: 6
TaskStatus: OK
...

PercentageComplete: 100
TaskState: Completed
PercentComplete: 100
TaskStatus: OK
```

(continues on next page)

(continued from previous page)

```
Firmware update successful!  
Overall Time Taken: 0:36:11  
Refer to 'DGX B200 Firmware Update Document' on activation steps for new firmware  
→to take effect.
```

3. Reset the BMC so the BMC boots the new firmware:

```
$ ipmitool mc reset cold  
  
# Wait a couple of minutes and confirm the BMC is back online.  
$ ipmitool mc info
```

4. Reboot the system.

Chapter 10. Firmware Update of Motherboard Tray: All Components

Perform the following steps to update the firmware on all the system components, such as CPLDs, PSUs, PCIe switches, and so on.

This procedure is an alternative to updating each component individually.

1. Create a `mb_tray.json` file with empty braces, like the following example:

```
{}
```

2. Update the firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-username> password=<bmc-password> update_
↪fw \
  -p nvfw_DGX_250220.1.0.fwpkg -y -s mb_tray.json
```

Example Output

```
FW package: ['nvfw_DGX_250220.1.0.fwpkg']
Ok to proceed with firmware update? <Y/N>
y
{"@odata.type": "#UpdateService.v1_11_0.UpdateService", "Messages": [{"@odata.type":
↪ "#Message.v1_0_8.Message", "Message": "A new task /redfish/v1/TaskService/
↪ Tasks/2 was created.", "MessageArgs": ["/redfish/v1/TaskService/Tasks/2"],
↪ "MessageId": "Task.1.0.New", "Resolution": "None", "Severity": "OK"}, {"@odata.
↪ type": "#Message.v1_0_8.Message", "Message": "The action UpdateService.
↪ MultipartPush was submitted to do firmware update.", "MessageArgs": [
↪ "UpdateService.MultipartPush"], "MessageId": "UpdateService.1.0.
↪ StartFirmwareUpdate", "Resolution": "None", "Severity": "OK"}]}
FW update started, Task Id: 2
Wait for Firmware Update to Start...
TaskState: Running
PercentComplete: 1
TaskStatus: OK
TaskState: Running
PercentComplete: 20
TaskStatus: OK
TaskState: Running
PercentComplete: 40
TaskStatus: OK
```

(continues on next page)

(continued from previous page)

```
TaskState: Running
PercentComplete: 61
TaskStatus: OK
TaskState: Running
PercentComplete: 80
TaskStatus: OK
TaskState: Running
PercentComplete: 99
TaskStatus: OK
TaskState: Completed
PercentComplete: 100
TaskStatus: OK
Firmware update successful!
Overall Time Taken: 0:24:38
Refer to 'DGX B200 Firmware Update Document' on activation steps for new firmware
↳to take effect.
-----
↳-----
```

Chapter 11. Updating BMC EROT on the Motherboard Tray

1. Create a `updparameters.json` file with the following contents:

```
{
  "Targets" : ["/redfish/v1/UpdateService/FirmwareInventory/EROT_BMC_0"]
}
```

2. Update the firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=admin password=admin update_fw \
-p nvfw_DGX_250220.1.0.fwpkg -y -s updparameters.json
```

Example Output

```
FW recipe: ['nvfw_DGX_250220.1.0.fwpkg']
{"@odata.type": "#UpdateService.v1_6_0.UpdateService", "Messages": [{"@odata.type": "#Message.v1_0_8.Message", "Message": "A new task /redfish/v1/TaskService/Tasks/1 was created.", "MessageArgs": ["/redfish/v1/TaskService/Tasks/1"], "MessageId": "Task.1.0.New", "Resolution": "None", "Severity": "OK"}, {"@odata.type": "#Message.v1_0_8.Message", "Message": "The action UpdateService.MultipartPush was submitted to do firmware update.", "MessageArgs": ["UpdateService.MultipartPush"], "MessageId": "UpdateService.1.0.StartFirmwareUpdate", "Resolution": "None", "Severity": "OK"}]}
FW update started, Task Id: 1
Wait for Firmware Update to Start...
Wait for Firmware Update to Start...
TaskState: Completed
PercentComplete: 100
TaskStatus: OK
Firmware update successful!
Overall Time Taken: 0:00:09
Refer to 'DGX B200 Firmware Update Document' on activation steps for new firmware
to take effect.
```

Chapter 12. Updating SBIOS EROT on the Motherboard Tray

1. Create a `updparameters.json` file with the following contents:

```
{
  "Targets" : ["/redfish/v1/UpdateService/FirmwareInventory/EROT_BIOS_0"]
}
```

2. Update the firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=admin password=admin update_fw \
-p nvfw_DGX_250220.1.0.fwpkg -y -s updparameters.json
```

Example Output

```
FW recipe: ['nvfw_DGX_250220.1.0.fwpkg.fwpkg']
{"@odata.type": "#UpdateService.v1_6_0.UpdateService", "Messages": [{"@odata.type": "#Message.v1_0_8.Message", "Message": "A new task /redfish/v1/TaskService/Tasks/2 was created.", "MessageArgs": ["/redfish/v1/TaskService/Tasks/2"], "MessageId": "Task.1.0.New", "Resolution": "None", "Severity": "OK"}, {"@odata.type": "#Message.v1_0_8.Message", "Message": "The action UpdateService.MultipartPush was submitted to do firmware update.", "MessageArgs": ["UpdateService.MultipartPush"], "MessageId": "UpdateService.1.0.StartFirmwareUpdate", "Resolution": "None", "Severity": "OK"}]}
FW update started, Task Id: 2
Wait for Firmware Update to Start...
Wait for Firmware Update to Start...
TaskState: Completed
PercentComplete: 100
TaskStatus: OK
Firmware update successful!
Overall Time Taken: 0:00:10
Refer to 'DGX B200 Firmware Update Document' on activation steps for new firmware
to take effect.
```

Chapter 13. Updating the BIOS on the Motherboard Tray

1. Create a `updparameters.json` file with the following contents:

```
{
  "Targets" : ["/redfish/v1/UpdateService/FirmwareInventory/HostBIOS_0"]
}
```

2. Update the firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=admin password=admin update_fw \
-p nvfw_DGX_250220.1.0.fwpkg -y -s updparameters.json
```

Example Output

```
FW recipe: [.1nvfw_DGX_250220.1.0.fwpkg']
{"@odata.type": "#UpdateService.v1_6_0.UpdateService", "Messages": [{"@odata.type": "#Message.v1_0_8.Message", "Message": "A new task /redfish/v1/TaskService/Tasks/2 was created.", "MessageArgs": ["/redfish/v1/TaskService/Tasks/2"], "MessageId": "Task.1.0.New", "Resolution": "None", "Severity": "OK"}, {"@odata.type": "#Message.v1_0_8.Message", "Message": "The action UpdateService.MultipartPush was submitted to do firmware update.", "MessageArgs": ["/redfish/v1/UpdateService.MultipartPush"], "MessageId": "UpdateService.1.0.StartFirmwareUpdate", "Resolution": "None", "Severity": "OK"}]}
FW update started, Task Id: 2
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
Wait for FirmwareUpdateStarted in MessageId
PercentageComplete: 6
TaskState: Running
PercentComplete: 6
TaskStatus: OK
...
PercentageComplete: 100
TaskState: Completed
PercentComplete: 100
```

(continues on next page)

(continued from previous page)

```
TaskStatus: OK
Firmware update successful!
Overall Time Taken: 0:36:11
Refer to 'DGX B200 Firmware Update Document' on activation steps for new firmware
↔to take effect.
```

Chapter 14. Updating the CPLDs on the Motherboard Tray

1. Create a `updparameters.json` file with the following contents:

```
{
  "Targets" : ["/redfish/v1/UpdateService/FirmwareInventory/CPLDMB_0"]
}
```

2. Update the firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=admin password=admin update_fw \
-p nvfw_DGX_250220.1.0.fwpkg -y -s updparameters.json
```

Example Output

```
FW recipe: ['nvfw_DGX_250220.1.0.fwpkg']
{"@odata.type": "#UpdateService.v1_6_0.UpdateService", "Messages": [{"@odata.type": "#Message.v1_0_8.Message", "Message": "A new task /redfish/v1/TaskService/Tasks/4 was created.", "MessageArgs": ["/redfish/v1/TaskService/Tasks/4"], "MessageId": "Task.1.0.New", "Resolution": "None", "Severity": "OK"}, {"@odata.type": "#Message.v1_0_8.Message", "Message": "The action UpdateService.MultipartPush was submitted to do firmware update.", "MessageArgs": ["UpdateService.MultipartPush"], "MessageId": "UpdateService.1.0.StartFirmwareUpdate", "Resolution": "None", "Severity": "OK"}]}
FW update started, Task Id: 4
Wait for Firmware Update to Start...
Wait for Firmware Update to Start...
TaskState: Completed
PercentComplete: 100
TaskStatus: OK
Firmware update successful!
Overall Time Taken: 0:00:08
Refer to 'DGX B200 Firmware Update Document' on activation steps for new firmware
to take effect.
```

Chapter 15. Updating the Midplane CPLDs on the Motherboard Tray

1. Create a `updparameters.json` file with the following contents:

```
{
  "Targets" : ["/redfish/v1/UpdateService/FirmwareInventory/CPLDMID_0"]
}
```

2. Update the firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=admin password=admin update_fw \
-p nvfw_DGX_250220.1.0.fwpkg -y -s updparameters.json
```

Example Output

```
FW recipe: ['nvfw_DGX_250220.1.0.fwpkg']
{"@odata.type": "#UpdateService.v1_6_0.UpdateService", "Messages": [{"@odata.type": "#Message.v1_0_8.Message", "Message": "A new task /redfish/v1/TaskService/Tasks/5 was created.", "MessageArgs": ["/redfish/v1/TaskService/Tasks/5"], "MessageId": "Task.1.0.New", "Resolution": "None", "Severity": "OK"}, {"@odata.type": "#Message.v1_0_8.Message", "Message": "The action UpdateService.MultipartPush was submitted to do firmware update.", "MessageArgs": ["UpdateService.MultipartPush"], "MessageId": "UpdateService.1.0.StartFirmwareUpdate", "Resolution": "None", "Severity": "OK"}]}
FW update started, Task Id: 5
Wait for Firmware Update to Start...
Wait for Firmware Update to Start...
  TaskState: Completed
  PercentComplete: 100
  TaskStatus: OK
Firmware update successful!
  Overall Time Taken: 0:00:09
Refer to 'DGX B200 Firmware Update Document' on activation steps for new firmware
↳ to take effect.
```

Chapter 16. Updating the NVMe Firmware

The following instructions describe how to update the firmware on an NVMe device using the NVME-CLI command-line interface. The `nvme-cli` tool is preinstalled on all DGX B200 systems.

1. List the devices and check the firmware versions.

```
$ sudo nvme list
```

| Node | Generic | SN | Model | FW Rev |
|-----------------------|------------|----------------|----------------------------|--------|
| ↳Namespace Usage | | | Format | |
| ↳ /dev/nvme0n1 | /dev/ng0n1 | S666NN0X307344 | SAMSUNG MZ1L21T9HCLS-00A07 | 0x1 |
| ↳ 589.06 TB / 1.92 TB | | 512 B + 0 B | GDC7502Q | |
| ↳ /dev/nvme1n1 | /dev/ng1n1 | S666NN0X307345 | SAMSUNG MZ1L21T9HCLS-00A07 | 0x1 |
| ↳ 589.74 TB / 1.92 TB | | 512 B + 0 B | GDC7502Q | |
| ↳ /dev/nvme2n1 | /dev/ng2n1 | YEE0A0750LS3 | KIOXIA KCMYDRUG3T84 | 0x1 |
| ↳ 247.45 GB / 3.84 TB | | 512 B + 0 B | 1UET7104 | |
| ↳ /dev/nvme3n1 | /dev/ng3n1 | YEC0A0XS0LS3 | KIOXIA KCMYDRUG3T84 | 0x1 |
| ↳ 175.18 GB / 3.84 TB | | 512 B + 0 B | 1UET7104 | |
| ↳ /dev/nvme4n1 | /dev/ng4n1 | 9EX0A0GU0LS3 | KIOXIA KCMYDRUG3T84 | 0x1 |
| ↳ 188.78 GB / 3.84 TB | | 512 B + 0 B | 1UET7104 | |
| ↳ /dev/nvme5n1 | /dev/ng5n1 | YEE0A0AL0LS3 | KIOXIA KCMYDRUG3T84 | 0x1 |
| ↳ 188.15 GB / 3.84 TB | | 512 B + 0 B | 1UET7104 | |
| ↳ /dev/nvme6n1 | /dev/ng6n1 | YEC0A0XC0LS3 | KIOXIA KCMYDRUG3T84 | 0x1 |
| ↳ 175.15 GB / 3.84 TB | | 512 B + 0 B | 1UET7104 | |
| ↳ /dev/nvme7n1 | /dev/ng7n1 | 9ET0A04T0LS3 | KIOXIA KCMYDRUG3T84 | 0x1 |
| ↳ 188.78 GB / 3.84 TB | | 512 B + 0 B | 1UET7104 | |
| ↳ /dev/nvme8n1 | /dev/ng8n1 | 9ET0A05B0LS3 | KIOXIA KCMYDRUG3T84 | 0x1 |
| ↳ 175.15 GB / 3.84 TB | | 512 B + 0 B | 1UET7104 | |
| ↳ /dev/nvme9n1 | /dev/ng9n1 | 9ET0A04X0LS3 | KIOXIA KCMYDRUG3T84 | 0x1 |
| ↳ 175.22 GB / 3.84 TB | | 512 B + 0 B | 1UET7104 | |

From the output, you can find the device names and firmware versions, such as `/dev/nvme0n1` and `GDC7502Q`.

2. Download the firmware you want to upgrade for the NVMe device.

Based on the command output in step 1, determine the firmware file using the following table and include the correct path to download the firmware:

```
$ sudo nvme fw-download $DeviceName --fw=***(specify the firmware file for  
↳upgrade)
```

This table lists the firmware file names for the supported Samsung and Kioxia models:

| Manufacturer | Model Number | Firmware File |
|--------------|--------------------|---|
| Samsung | MZ1L21T9HCLS-00A07 | General_PM9A3_M. 2_GDC7502Q_Noformat.bin |
| Samsung | MZWLO3T8HCLS-00A07 | General_PM1743_U.2_OPPA4B5Q.bin |
| Kioxia | KCM6DRUL3T84 | CM6-SED-0107.std |
| Kioxia | KCM7DRUL3T84 | CM6-SED-0107.std |

3. Commit and activate the downloaded firmware immediately without reset by setting the action argument to 3.

```
$ sudo nvme fw-commit $DeviceName --action=3
```

4. Verify that the correct firmware version is updated.

```
$ sudo nvme id-ctrl $DeviceName
```

Chapter 17. Updating the Power Supply Units on the Motherboard Tray

1. Create a `updparameters.json` file with the following contents:

```
{
  "Targets" : ["/redfish/v1/UpdateService/FirmwareInventory/PSU_0"]
}
```

Important

Repeat this procedure for PSU_1 through PSU_5.

2. Update the firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=admin password=admin update_fw \
-p nvfw_DGX_250220.1.0.fwpkg -y -s updparameters.json
```

Example Output

```
FW recipe: ['nvfw_DGX_250220.1.0.fwpkg']
{"@odata.type": "#UpdateService.v1_6_0.UpdateService", "Messages": [{"@odata.type": "#Message.v1_0_8.Message", "Message": "A new task /redfish/v1/TaskService/Tasks/6 was created.", "MessageArgs": ["/redfish/v1/TaskService/Tasks/6"], "MessageId": "Task.1.0.New", "Resolution": "None", "Severity": "OK"}, {"@odata.type": "#Message.v1_0_8.Message", "Message": "The action UpdateService.MultipartPush was submitted to do firmware update.", "MessageArgs": ["UpdateService.MultipartPush"], "MessageId": "UpdateService.1.0.StartFirmwareUpdate", "Resolution": "None", "Severity": "OK"}]}
FW update started, Task Id: 6
Wait for Firmware Update to Start...
Wait for Firmware Update to Start...
TaskState: Completed
PercentComplete: 100
TaskStatus: OK
Firmware update successful!
Overall Time Taken: 0:00:08
Refer to 'DGX B200 Firmware Update Document' on activation steps for new firmware
to take effect.
```

Chapter 18. Updating the PCIe Switches on the Motherboard Tray

1. Create a `updparameters.json` file with the following contents:

```
{
  "Targets" : ["/redfish/v1/UpdateService/FirmwareInventory/PCIESwitch_0"]
}
```

Important

Repeat this procedure for `PCIESwitch_1`.

2. Update the firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=admin password=admin update_fw \
-p nvfw_DGX_250220.1.0.fwpkg -y -s updparameters.json
```

Example Output

```
FW recipe: ['nvfw_DGX_250220.1.0.fwpkg']
{"@odata.type": "#UpdateService.v1_6_0.UpdateService", "Messages": [{"@odata.type": "#Message.v1_0_8.Message", "Message": "A new task /redfish/v1/TaskService/Tasks/7 was created.", "MessageArgs": ["/redfish/v1/TaskService/Tasks/7"], "MessageId": "Task.1.0.New", "Resolution": "None", "Severity": "OK"}, {"@odata.type": "#Message.v1_0_8.Message", "Message": "The action UpdateService.MultipartPush was submitted to do firmware update.", "MessageArgs": ["UpdateService.MultipartPush"], "MessageId": "UpdateService.1.0.StartFirmwareUpdate", "Resolution": "None", "Severity": "OK"}]}
FW update started, Task Id: 7
Wait for Firmware Update to Start...
Wait for Firmware Update to Start...
TaskState: Completed
PercentComplete: 100
TaskStatus: OK
Firmware update successful!
Overall Time Taken: 0:00:09
Refer to 'DGX B200 Firmware Update Document' on activation steps for new firmware
to take effect.
```

Chapter 19. Updating the PCIe Retimers on the Motherboard Tray

1. Create a `updparameters.json` file with the following contents:

```
{
  "Targets" : ["/redfish/v1/UpdateService/FirmwareInventory/PCIeRetimer_0"]
}
```

Important

Repeat this procedure for `PCIeRetimer_1`.

2. Update the firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=admin password=admin update_fw \
-p nvfw_DGX_250220.1.0.fwpkg -y -s updparameters.json
```

Example Output

```
FW recipe: ['nvfw_DGX_250220.1.0.fwpkg']
{"@odata.type": "#UpdateService.v1_6_0.UpdateService", "Messages": [{"@odata.type": "#Message.v1_0_8.Message", "Message": "A new task /redfish/v1/TaskService/Tasks/8 was created.", "MessageArgs": ["/redfish/v1/TaskService/Tasks/8"], "MessageId": "Task.1.0.New", "Resolution": "None", "Severity": "OK"}, {"@odata.type": "#Message.v1_0_8.Message", "Message": "The action UpdateService.MultipartPush was submitted to do firmware update.", "MessageArgs": ["/redfish/v1/UpdateService.MultipartPush"], "MessageId": "UpdateService.1.0.StartFirmwareUpdate", "Resolution": "None", "Severity": "OK"}]}
FW update started, Task Id: 8
Wait for Firmware Update to Start...
Wait for Firmware Update to Start...
TaskState: Completed
PercentComplete: 100
TaskStatus: OK
Firmware update successful!
Overall Time Taken: 0:00:09
Refer to 'DGX B200 Firmware Update Document' on activation steps for new firmware
to take effect.
```

Chapter 20. Updating the ConnectX-7 Firmware

After replacing or installing the ConnectX-7 cards, make sure the firmware on the cards is up to date. Refer to the [component firmware versions table](#) to find the most recent firmware version.

1. Download the firmware from <https://network.nvidia.com/support/firmware/connectx7ib/>.
Download the firmware for both OPN options.
2. Transfer the firmware ZIP file to the DGX system and extract the archive.
3. Update the firmware on the cards that are used for cluster communication:

```
sudo mlxfwmanager -d /sys/bus/pci/devices/0000:5e:00.0/config -i fw-ConnectX7-rel-
↪28_43_2026-MCX750500B-0D00_Ax_Bx-UEFI-14.36.21-FlexBoot-3.7.500.signed.bin b
sudo mlxfwmanager -d /sys/bus/pci/devices/0000:dc:00.0/config -i fw-ConnectX7-rel-
↪28_43_2026-MCX750500B-0D00_Ax_Bx-UEFI-14.36.21-FlexBoot-3.7.500.signed.bin b
sudo mlxfwmanager -d /sys/bus/pci/devices/0000:c0:00.0/config -i fw-ConnectX7-rel-
↪28_43_2026-MCX750500B-0D00_Ax_Bx-UEFI-14.36.21-FlexBoot-3.7.500.signed.bin b
sudo mlxfwmanager -d /sys/bus/pci/devices/0000:18:00.0/config -i fw-ConnectX7-rel-
↪28_43_2026-MCX750500B-0D00_Ax_Bx-UEFI-14.36.21-FlexBoot-3.7.500.signed.bin b
sudo mlxfwmanager -d /sys/bus/pci/devices/0000:40:00.0/config -i fw-ConnectX7-rel-
↪28_43_2026-MCX750500B-0D00_Ax_Bx-UEFI-14.36.21-FlexBoot-3.7.500.signed.bin b
sudo mlxfwmanager -d /sys/bus/pci/devices/0000:4f:00.0/config -i fw-ConnectX7-rel-
↪28_43_2026-MCX750500B-0D00_Ax_Bx-UEFI-14.36.21-FlexBoot-3.7.500.signed.bin b
sudo mlxfwmanager -d /sys/bus/pci/devices/0000:ce:00.0/config -i fw-ConnectX7-rel-
↪28_43_2026-MCX750500B-0D00_Ax_Bx-UEFI-14.36.21-FlexBoot-3.7.500.signed.bin b
sudo mlxfwmanager -d /sys/bus/pci/devices/0000:9a:00.0/config -i fw-ConnectX7-rel-
↪28_43_2026-MCX750500B-0D00_Ax_Bx-UEFI-14.36.21-FlexBoot-3.7.500.signed.bin b
```

4. Update the firmware on the cards that are used for storage communication:

```
sudo mlxfwmanager -d /sys/bus/pci/devices/0000:aa:00.0/config -i fw-ConnectX7-rel-
↪28_43_2026-MCX755206AS-NEA_Ax-UEFI-14.36.21-FlexBoot-3.7.500.signed.bin b
sudo mlxfwmanager -d /sys/bus/pci/devices/0000:29:00.0/config -i fw-ConnectX7-rel-
↪28_43_2026-MCX755206AS-NEA_Ax-UEFI-14.36.21-FlexBoot-3.7.500.signed.bin b
```

5. Perform an AC power cycle on the system for the firmware update to take effect.
Wait for the operating system to boot.
6. After the system starts, log in and confirm the firmware versions are all the same:

```
$ cat /sys/class/infiniband/mlx5_*/fw_ver
```

Chapter 21. Updating the Intel NIC Firmware

The following instructions describe how to update the firmware on the Intel Ethernet Network Adapter E810-C using interactive mode.

1. Download the update package to a temporary directory.
2. Start the update by running the Intel Ethernet NVM Update Tool `nvmupdate64e`.

```
sudo ./nvmupdate64e
```

3. Follow the prompts to update the NVM image on the network adapter.

Example output:

```
$ sudo ./nvmupdate64e

Intel(R) Ethernet NVM Update Tool
NVMUpdate version 1.41.3.3
Copyright(C) 2013 - 2024 Intel Corporation.

WARNING: To avoid damage to your device, do not stop the update or reboot or
↪power off the system during this update.
Inventory in progress. Please wait [***.....]

Num Description                               Ver.(hex)  DevId S:B   Status
=== =====
01) Intel(R) Ethernet Controller             N/A(N/A)   1563 00:011 Update not
     X550-T2                                  available
02) Intel(R) Ethernet Network Adapter        2.80(2.50) 1592 00:130 Update
     E810-C-Q2                                  available

Options: Adapter Index List (comma-separated), [A]ll, e[X]it
Enter selection: 02
Would you like to back up the NVM images? [Y]es/[N]o: Y
Update in progress. This operation may take several minutes.
[*+.....]

Num Description                               Ver.(hex)  DevId S:B   Status
=== =====
01) Intel(R) Ethernet Controller             N/A(N/A)   1563 00:011 Update not
```

(continues on next page)

(continued from previous page)

```
X550-T2                                available
02) Intel(R) Ethernet Network Adapter  2.80(2.50)  1592 00:130 Update
E810-C-Q2                                successful
```

A reboot is required to complete the update process.

```
Tool execution completed with the following status: All operations completed
↪successfully.
Press any key to exit.
```

4. Perform an AC power cycle on the system for the firmware update to take effect.
Wait for the operating system to boot.

Chapter 22. Firmware Update of GPU Tray: All Components

Perform the following steps to update the firmware on all the components in the GPU tray, such as GPUs, NVSwitches, and so on.

1. Create a `gpu_tray.json` file with the following contents:

```
{
  "Targets" : ["/redfish/v1/UpdateService/FirmwareInventory/HGX_0"]
}
```

2. Update the firmware:

```
nvfwupd -t ip=<bmc-ip-address> user=<bmc-username> password=<bmc-password> update_
↪fw \
  -p nvfw_DGX-HGX-B100-B200x8_250124.1.3.fwpkg -y -s gpu_tray.json
```

Example Output

```
FW recipe: ['HGX_DGXB100-B200x8_250124.1.3.fwpkg']
{"@odata.type": "#UpdateService.v1_6_0.UpdateService", "Messages": [{"@odata.type
↪": "#Message.v1_0_8.Message", "Message": "A new task /redfish/v1/TaskService/
↪Tasks/3 was created.", "MessageArgs": ["/redfish/v1/TaskService/Tasks/3"],
↪"MessageId": "Task.1.0.New", "Resolution": "None", "Severity": "OK"}, {"@odata.
↪type": "#Message.v1_0_8.Message", "Message": "The action UpdateService.
↪MultipartPush was submitted to do firmware update.", "MessageArgs": [
↪"UpdateService.MultipartPush"], "MessageId": "UpdateService.1.0.
↪StartFirmwareUpdate", "Resolution": "None", "Severity": "OK"}]}
FW update started, Task Id: 3
Wait for Firmware Update to Start...
Wait for Firmware Update to Start...
Started Updating: HGX_0
TaskState: Running
PercentComplete: 20
TaskStatus: OK
TaskState: Running
PercentComplete: 40
TaskStatus: OK
TaskState: Completed
PercentComplete: 100
TaskStatus: OK
Firmware update successful!
Overall Time Taken: 0:09:14
Refer to
```

Chapter 23. Platform-Agnostic Firmware Updates

Note

Deprecated feature: Starting with nvfwupd version 2.0.5, specifying the `targets` sub-option with a JSON file of multiple targets for multiple system updates is no longer supported.

The nvfwupd versions 2.0.0 and later support input and platform-agnostic firmware updates through the YAML configuration file. You can customize update methods and the Redfish URIs used for the update.

23.1. Configuration Parameters

The tool supports various configuration parameters, of which `BMC_IP`, `RF_USERNAME`, and `RF_PASSWORD` are always mandatory. The following example YAML file explains each parameter:

```
# Define target platform as one of HGX / DGX / GH / NVOS / GH200
TargetPlatform: 'GH'
# Disable Sanitize Log, disabling Sanitize Log leads to print system IP and user
↳ credential to the logs and screen
SANITIZE_LOG: False

# Provide full path of firmware file(s) to be used for firmware update. Value is a list
FWUpdateFilePath:
- "../packages/ nvfw_Grace-CPU-P5041_0003_240124.1.0_prod-signed.fwpkg"

# Define URI for MultipartHttpPushUri (Optional override).
# Default value is taken from UpdateService
MultipartHttpPushUri: '/redfish/v1/UpdateService/update-multipart'

# HttpPushUri for PLDM Firmware Update (Optional override).
# Default value "/redfish/v1/UpdateService"
HttpPushUri: '/redfish/v1/UpdateService'

# Change if TaskServiceUri is different from default value below
# Task id will always be taken from update response in update_fw or
# from -i input in show_update_progress
TaskServiceUri: '/redfish/v1/TaskService/Tasks/'
```

(continues on next page)

```
# Define differnt update methods.
# Valid values {'MultipartHttpPushUri', 'HttpPushUri', 'NVOSRest'}
FwUpdateMethod: "MultipartHttpPushUri"

# Optional Parameter used with MultipartHttpPushUri update method
# used to define dict of parameters for multipart FW update
MultipartOptions:
- ForceUpdate: True

# Target IP address. BMC IP/NVOS Rest service IP/localhost for port forwarding
BMC_IP: "1.1.1.1"
RF_USERNAME: "user"
RF_PASSWORD: "password"
# Target port config if port forwarding is used.
TUNNEL_TCP_PORT: "14443"

# List of update targets. replaces -s/--special option input file. Value is list of
↪target URIs
# Use UpdateParametersTargets: {} for DGX empty JSON value used for full DGX update
# Use UpdateParametersTargets: [] for Oberon empty list used for BMC update
# Remove this parameter for GH full update
UpdateParametersTargets:
- "/redfish/v1/UpdateService/FirmwareInventory/CPLDMB_0"

# Config for reset BMC parameters. Value is a dict.
# Use ResetType: 'ResetAll' for DGX
# Use ResetToDefaultsType: 'ResetAll' for MGX or HGX
BMCResetParameters:
  ResetType: 'ResetAll'

# Multi target input. Value is list of dicts.
Targets:
- BMC_IP: "1.1.1.1"
  RF_USERNAME: "user"
  RF_PASSWORD: "password"
  TUNNEL_TCP_PORT: "14443"
- BMC_IP: "2.2.2.2"
  RF_USERNAME: "user"
  RF_PASSWORD: "password"
  TUNNEL_TCP_PORT: "14444"
```

23.2. Running nvfwupd Using the Configuration File

To use the tool and update a platform that supports MultipartHttpPushUri or HttpPushUri, but is not automatically identified by the tool or provides a platform that is not a supported error, a configuration file can be used to provide the input and customize the behavior.

Keep the following points in mind:

- ▶ Support for show_version on an unknown platform is limited.

If the `TargetPlatform` parameter is not in the configuration file, `show_version` will not match the firmware inventory to PLDM package contents. The **Pkg Version** and **Up-to-date** columns will show N/A and No, respectively.

- ▶ The `make_upd_targets` command is not supported because the resulting JSON files cannot be used with the configuration file.
- ▶ The configuration file takes update targets as a configuration parameter, and because the tool is supposed to be used with a platform that is not known to the tool, you must identify and verify the target list before providing it as input.

Examples:

```
$ cat config.yaml
TargetPlatform: 'GH'
FWUpdateFilePath:
- "../packages/nvfw_P4764_0000_240103.1.2_dbg-signed.fwpkg"
MultipartHttpPushUri: '/redfish/v1/UpdateService/update-multipart'
FwUpdateMethod: 'MultipartHttpPushUri'
BMC_IP: "1.1.1.1"
RF_USERNAME: "****"
RF_PASSWORD: "*****"
UpdateParametersTargets:
- "/redfish/v1/Chassis/5B247A_Baseboard_0"
BMCResetParameters:
ResetToDefaultsType: 'ResetAll'
```

```
$ nvfwupd -c config.yaml update_fw
Updating ip address: ip=1.1.1.1
FW package: ['../packages/nvfw_P4764_0000_240103.1.2_dbg-signed.fwpkg']
Ok to proceed with firmware update? <Y/N>
y
{
  "@odata.id": "/redfish/v1/TaskService/Tasks/5B247A_5",
  "@odata.type": "#Task.v1_4_3.Task",
  "Id": "5B247A_5",
  "TaskState": "Running",
  "TaskStatus": "OK"
}
FW update started, Task Id: 5B247A_5
Wait for Firmware Update to Start...
TaskState: Running
PercentComplete: 20
TaskStatus: OK
TaskState: Running
PercentComplete: 40
TaskStatus: OK
TaskState: Completed
PercentComplete: 100
TaskStatus: OK
Firmware update successful!
Overall Time Taken: 0:08:15
Update successful. Perform activation steps for new firmware to take effect.
-----
↩-
Error Code: 0
```

```
$ nvfwupd -c oregon_config.yaml perform_factory_reset

BMC IP: ip=1.1.1.1
Factory Reset request successful
Task State:
{
  "@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"
    }
  ]
}
-----
↩ -
```

```
$ nvfwupd -c config.yaml show_update_progress -i 0

Task Info for Id: 0
StartTime: 2024-01-20T02:46:15+00:00
TaskState: Completed
PercentComplete: 100
TaskStatus: OK
EndTime: 2024-01-20T02:46:17+00:00
Overall Time Taken: 0:00:02
Overall Task Status: {
  "@odata.id": "/redfish/v1/TaskService/Tasks/0",
  "@odata.type": "#Task.v1_4_3.Task",
  "EndTime": "2024-01-20T02:46:17+00:00",
  "Id": "0",
  "Messages": [
    {
      "@odata.type": "#Message.v1_0_0.Message",
      "Message": "The task with id 0 has started.",
      "MessageArgs": [
        "0"
      ],
      "MessageId": "TaskEvent.1.0.1.TaskStarted",
      "Resolution": "None.",
      "Severity": "OK"
    },
    {
      "@odata.type": "#Message.v1_0_0.Message",
      "Message": "↩complete.",
      "MessageArgs": [
        "0",
        "100"
      ],
      "MessageId": "TaskEvent.1.0.1.TaskProgressChanged",
      "Resolution": "None.",
      "Severity": "OK"
    }
  ],
}
```

(continues on next page)

(continued from previous page)

```
{
  "@odata.type": "#Message.v1_0_0.Message",
  "Message": "The task with id 0 has Completed.",
  "MessageArgs": [
    "0"
  ],
  "MessageId": "TaskEvent.1.0.1.TaskCompletedOK",
  "Resolution": "None.",
  "Severity": "OK"
}
],
"Name": "Task 0",
"Payload": {
  "HttpHeaders": [
    "Host: 1.1.1.1",
    "User-Agent: python-requests/2.28.2",
    "Accept-Encoding: gzip, deflate",
    "Accept: */*",
    "Connection: keep-alive",
    "Content-Length: 109023143"
  ],
  "HttpOperation": "POST",
  "JsonBody": "null",
  "TargetUri": "/redfish/v1/UpdateService/update-multipart"
},
"PercentComplete": 100,
"StartTime": "2024-01-20T02:46:15+00:00",
"TaskMonitor": "/redfish/v1/TaskService/Tasks/0/Monitor",
"TaskState": "Completed",
"TaskStatus": "OK"
}
Update is successful.
-----
↩ -
```

Chapter 24. Performing a Factory Reset

- ▶ Reset the DGX system firmware to factory default values:

```
nvfwupd --target ip=<bmc-ip-address> user=admin password=admin perform_factory_
↪reset
```

Example Output

```
Factory Reset request successful
Task State:
{"@odata.context": "/redfish/v1/$metadata#Task.Task", "@odata.id": "/redfish/v1/
↪TaskService/Tasks/2", "@odata.type": "#Task.v1_4_2.Task", "Description": "Task
↪for Manager ResetToDefaults", "Id": "2", "Name": "Manager ResetToDefaults",
↪"TaskState": "New"}
```

Chapter 25. Forcing a Firmware Downgrade

25.1. Prerequisites

- ▶ Refer to *Viewing the Installed Firmware and Package Versions* to confirm that the firmware package has the firmware version that you want.

25.2. Procedure

1. Enable the ForceUpdate flag on the BMC:

```
nvfwupd --target ip=<bmc-ip-address> user=admin password=admin force_update enable
```

Example Output

```
ForceUpdate flag was successfully set True on the system.
```

2. Confirm the ForceUpdate flag status:

```
nvfwupd --target ip=<bmc-ip-address> user=admin password=admin force_update status
```

Example Output

```
ForceUpdate is set to True
```

3. Perform the firmware update.
4. Disable the ForceUpdate flag on the BMC:

```
nvfwupd --target ip=<bmc-ip-address> user=admin password=admin force_update  
↪disable
```

Example Output

```
ForceUpdate flag was successfully set False on the system.
```

5. Confirm the ForceUpdate flag status:

```
nvfwupd --target ip=<bmc-ip-address> user=admin password=admin force_update status
```

Example Output

```
ForceUpdate is set to False
```

Chapter 26. Troubleshooting an Unsuccessful Firmware Update

26.1. Firmware Update Terminates due to Component Not Found

When performing a firmware update of the GPU tray with the motherboard firmware package, the firmware update stops with the following output message:

```
...
{
  "@odata.type": "#Message.v1_0_8.Message",
  "Message": "Given PLDMBundle Status Message : Requested component was not found in
↳ the firmware bundle.",
  "MessageArgs": [
    "Requested component was not found in the firmware bundle."
  ],
  "MessageId": "UpdateService.1.0.FwUpdateStatusMessage",
  "Resolution": "None",
  "Severity": "Warning"
},
...
```

The message indicates that the firmware file specified by the `-p` argument of the `nvfwupd` command is invalid. Retry the update and specify the firmware file that matches the component. For example, use the GPU firmware file, which contains the HGX string, for the GPU tray update. Refer to [Version 25.01.1](#) for the firmware file names and components.

26.2. No Devices Where Detected for Handle ID 0

When performing a firmware update with the Redfish API, the following output message indicates that the firmware file specified in the `-F UpdateFile=` argument is not the correct file for the component specified in the JSON file.

```
...
{
  "@odata.type": "#Message.v1_0_8.Message",
  "Message": "Given PLDMBundle Status Message : No devices where detected for handle
↪ id 0.",
  "MessageArgs": [
    "No devices where detected for handle id 0"
  ],
  "MessageId": "UpdateService.1.0.FwUpdateStatusMessage",
  "Resolution": "None",
  "Severity": "Warning"
},
...
```

Retry the update and specify the firmware file that matches the component. Refer to [Redfish APIs Support](#) in the *NVIDIA DGX B200 System User Guide* for information about using the Redfish API.

26.3. Wait for Firmware Update Started ID

The output for an unsuccessful firmware update using the `nvfwupd` command can look like the following example:

```
FW recipe: ['nvfw_DGXB200_xxxx_xxxxxx.x.x.fwpkg']
{"@odata.type": "#UpdateService.v1_6_0.UpdateService", "Messages": [{"@odata.type": "
↪ #Message.v1_0_8.Message", "Message": "A new task /redfish/v1/TaskService/Tasks/4
↪ was created.", "MessageArgs": ["/redfish/v1/TaskService/Tasks/4"], "MessageId":
↪ "Task.1.0.New", "Resolution": "None", "Severity": "OK"}, {"@odata.type": "#Message.
↪ v1_0_8.Message", "Message": "The action UpdateService.MultipartPush was submitted
↪ to do firmware update.", "MessageArgs": ["UpdateService.MultipartPush"], "MessageId
↪ ": "UpdateService.1.0.StartFirmwareUpdate", "Resolution": "None", "Severity": "OK"}
↪ ]}
FW update started, Task Id: 4

Wait for FirmwareUpdateStarted Id in Messages
Wait for FirmwareUpdateStarted Id in Messages
Task Message: Task /redfish/v1/UpdateService/upload has stopped due to an exception
↪ condition.
Firmware update failed, retry the firmware update
```

Retry the firmware update, as indicated in the command output.

Chapter 27. Version 25.02.5

27.1. Highlights

27.1.1. Added Support

- ▶ Introduces support for the NVIDIA DGX™ B200 System.

27.1.2. BMC Fixes

- ▶ No BMC fixes for the initial release.

27.1.3. SBIOS Fixes

- ▶ No SBIOS fixes for the initial release.

27.1.4. The nvfwupd Command Updates

- ▶ Added support for parallel firmware updates through the YAML configuration file.
- ▶ Added the `--json` option to the `update_fw`, `show_update_progress`, and `force_update` commands.
- ▶ Added IPv6 support.
- ▶ Deprecated the `targets` sub-option for multi-target input. Use `config.yaml` input instead.

27.2. Firmware Package Details

This firmware release supports the following systems:

- ▶ NVIDIA DGX B200

This firmware release supports the following operating systems:

- ▶ NVIDIA DGX OS 7.0.1 and higher

For more information about the operating systems, refer to the [NVIDIA Base OS](#) documentation.

You can download firmware packages from the [NVIDIA Enterprise Support Portal](#).

The following table shows the firmware package files:

| Components | Sample File Name |
|--|---|
| Combined archive | DGXB200_25.02.5.tar.gz The combined archive includes the firmware for the system components and the firmware for the GPU tray. |
| <ul style="list-style-type: none"> ▶ Motherboard tray package ▶ GPU tray transition package ▶ GPU tray latest package | <ul style="list-style-type: none"> ▶ nvfw_DGX_250212.1.1.fwpkg ▶ nvfw_DGX-HGX-B100-B200x8_250114.1.0.fwpkg ▶ nvfw_DGX-HGX-B100-B200x8_250124.1.3.fwpkg |

The following table shows the information about component firmware versions.

| Component | Version |
|---|--|
| Host BMC | 25.02.12 |
| Host BMC ERoT | 04.0058 |
| SBIOS ERoT | 04.0058 |
| SBIOS | 1.6.6 |
| Motherboard CPLD | 0.2.1.9 |
| Midplane CPLD | 0.2.1.3 |
| PSU (Delta ECD16020137) | Primary 0204 Secondary 0201 Community 0204 |
| LiteOn | 0.5.0.5 |
| Broadcom Gen5 PCIe Switch (PEX89072-B01) | Switch 0: 0.0.7 Switch 1: 1.0.7 |
| | 2.07.19 |
| Astera Labs Gen5 PCIe Retimer (PT5161L) | |
| Network (Cluster) Card - ConnectX-7 | 28.43.2026 |
| Network (Storage) Card - ConnectX-7 | 28.43.2026 |
| Network Card - BlueField-3 | 32.43.2024 |
| VBIOS | 97.00.7F.00.01 |
| NVSwitch (GPU Tray) | 35.2014.1638 |
| ERoT (GPU Tray) | 02.0192 |
| HMC (GPU Tray) | HGX-24.11-1-rc57 |
| FPGA (GPU Tray) | 1.61 |
| PCIe Switch (GPU Tray) | 1.9.5F |
| | 2.7.20 |
| Astera Labs Gen5 PCIe Retimer (GPU Tray) (PT5161L) | |
| Intel 10G Ethernet | v3.60 |
| | v4.50 |

27.3. Firmware Update Procedure

Refer to *Firmware Update Steps*.

Chapter 28. Third-Party License Notices

This NVIDIA product contains third party software that is being made available to you under their respective open source software licenses. Some of those licenses also require specific legal information to be included in the product. This section provides such information.

28.1. Micron msecli

The `msecli` utility is provided under the following terms:

Micron Technology, Inc. Software License Agreement PLEASE READ THIS LICENSE AGREEMENT (“AGREEMENT”) FROM MICRON TECHNOLOGY, INC. (“MTI”) CAREFULLY: BY INSTALLING, COPYING OR OTHERWISE USING THIS SOFTWARE AND ANY RELATED PRINTED MATERIALS (“SOFTWARE”), YOU ARE ACCEPTING AND AGREEING TO THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE WITH THE TERMS OF THIS AGREEMENT, DO NOT INSTALL THE SOFTWARE. LICENSE: MTI hereby grants to you the following rights: You may use and make one (1) backup copy the Software subject to the terms of this Agreement. You must maintain all copyright notices on all copies of the Software. You agree not to modify, adapt, decompile, reverse engineer, disassemble, or otherwise translate the Software. MTI may make changes to the Software at any time without notice to you. In addition MTI is under no obligation whatsoever to update, maintain, or provide new versions or other support for the Software. OWNERSHIP OF MATERIALS: You acknowledge and agree that the Software is proprietary property of MTI (and/or its licensors) and is protected by United States copyright law and international treaty provisions. Except as expressly provided herein, MTI does not grant any express or implied right to you under any patents, copyrights, trademarks, or trade secret information. You further acknowledge and agree that all right, title, and interest in and to the Software, including associated proprietary rights, are and shall remain with MTI (and/or its licensors). This Agreement does not convey to you an interest in or to the Software, but only a limited right to use and copy the Software in accordance with the terms of this Agreement. The Software is licensed to you and not sold.

DISCLAIMER OF WARRANTY:

THE SOFTWARE IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND. MTI EXPRESSLY DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, NON-INFRINGEMENT OF THIRD PARTY RIGHTS, AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. MTI DOES NOT WARRANT THAT THE SOFTWARE WILL MEET YOUR REQUIREMENTS, OR THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE. FURTHERMORE, MTI DOES NOT MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. THE ENTIRE RISK ARISING OUT OF USE OR PERFORMANCE OF THE SOFTWARE REMAINS WITH YOU. IN NO EVENT SHALL MTI, ITS AFFILIATED COMPANIES OR THEIR SUPPLIERS BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL, OR SPECIAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF

INFORMATION) ARISING OUT OF YOUR USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF MTI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Because some jurisdictions prohibit the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

TERMINATION OF THIS LICENSE: MTI may terminate this license at any time if you are in breach of any of the terms of this Agreement. Upon termination, you will immediately destroy all copies the Software.

GENERAL: This Agreement constitutes the entire agreement between MTI and you regarding the subject matter hereof and supersedes all previous oral or written communications between the parties. This Agreement shall be governed by the laws of the State of Idaho without regard to its conflict of laws rules.

CONTACT: If you have any questions about the terms of this Agreement, please contact MTI's legal department at (208) 368-4500. By proceeding with the installation of the Software, you agree to the terms of this Agreement. You must agree to the terms in order to install and use the Software.

28.2. Mellanox (OFED)

MLNX_OFED <<http://www.mellanox.com/>> is provided under the following terms:

Copyright (c) 2006 Mellanox Technologies. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Chapter 29. Notices

29.1. 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 NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

29.2. Trademarks

NVIDIA, the NVIDIA logo, DGX, DGX-1, DGX-2, DGX A100, DGX H100/H200, DGX B200, DGX Station, and DGX Station A100 are trademarks and/or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

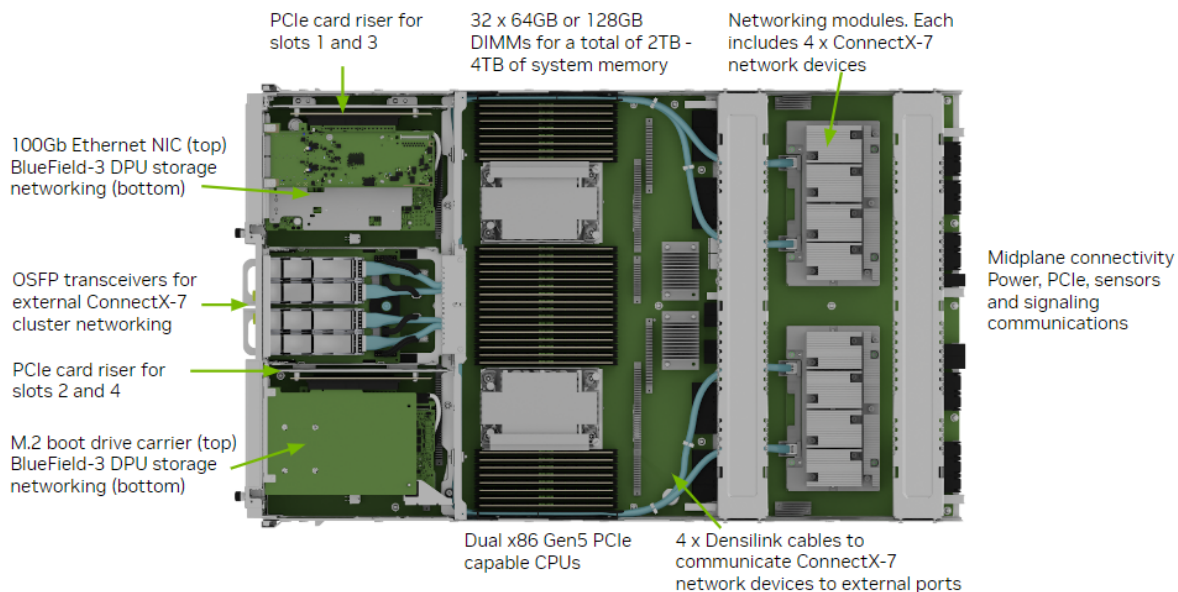
Chapter 30. About Firmware Updates

This topic provides an overview of firmware updates on the NVIDIA DGX™ B200 system.

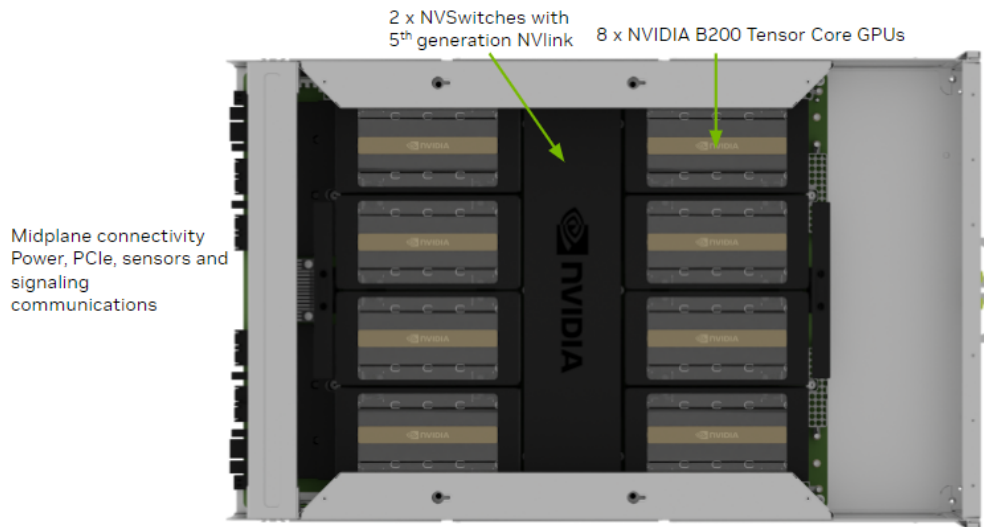
30.1. Firmware Updatable Components

The NVIDIA DGX™ B200 system has several firmware updatable components. Some of the components are on the following two trays in the system:

- ▶ The motherboard tray has components, such as the CPUs, PCH, BMC as shown in the following figure:



- ▶ The GPU tray has components, such as the GPUs, NVSwitches, HMC as shown in the following figure:



You can update the firmware on the NVIDIA DGX B200 system components out-of-band (OOB) by using Redfish APIs or from the host operating system by using the command-line interface (CLI) commands.

30.2. Firmware Update Prerequisites

- ▶ You can download firmware packages from the [NVIDIA Enterprise Support Portal](#).
- ▶ You must know the BMC IP address, a user name, and a password. The sample commands in this document show `admin` for both the user name and the password.
- ▶ You must have the `nvfwupd` executable or know how to use the Redfish API.

30.3. Firmware Update Methods

Most of the sample commands in this document show how to update firmware using the `nvfwupd` command. You can download the executable from the [NVIDIA Enterprise Support Portal](#).

- ▶ For more information about the command, refer to [About the nvfwupd Command](#).
- ▶ For best practice when updating the firmware, follow the instructions in [Firmware Update Steps](#).

You can run the `nvfwupd` command interactively to update systems. Most command examples in this document show this interactive approach. If you have several systems to update, you can create a JSON file that identifies the systems to update. Refer to [Platform-Agnostic Firmware Updates](#) for more information.

An alternative to the `nvfwupd` command is to update firmware by using the Redfish API. The BMC network interface provides remote management with Redfish APIs.

The [Known Issues](#) for updating firmware and the [firmware update steps](#) still apply when you use the Redfish API.

Refer to [Redfish APIs Support](#) in the *NVIDIA DGX B200 System User Guide* for more information and sample commands. The sample commands show how to update firmware with the `curl` command.

30.4. Firmware Update Activation

After the firmware update, you must perform one or more of the following tasks to activate the firmware update, depending on the components being updated:

- ▶ BMC component

Reset the BMC by running the following command:

```
sudo ipmitool mc reset cold
```

- ▶ PCIe Switch, PCIe Retimer, BIOS, and HGX (GPU Tray) components

Perform a cold reset on the system using the following command:

```
sudo ipmitool chassis power cycle
```

- ▶ EROT and CPLD components

Perform an AC power cycle on the system by unplugging all the power supplies and then reconnecting them either manually or through an external PDU device.

Note

The AC power cycle will activate firmware for all updated components.

Copyright

©2022-2025, NVIDIA Corporation