NVIDIA DOCA

Installation Guide

MLNX-15-060470 _v1.2 | January 2022

Table of Contents

Chapter 1. Introduction

There are two ways to install the NVIDIA BlueField-2 DPU software:

- Using the SDK Manager which provides a GUI/CLI for full BlueField-2 installation
- Manual installation with a step-by-step procedure

1.1. Supported Platforms

Model Number	Description
MBF2H322A-AEEOT	NVIDIA® BlueField®-2 P-Series DPU 25GbE Dual-Port SFP56, PCIe Gen4 x8, Crypto Enabled, 8GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHL
MBF2H322A-AENOT	NVIDIA BlueField-2 P-Series DPU 25GbE Dual-Port SFP56, PCIe Gen4 x8, Crypto Disabled, 8GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHL
MBF2H332A-AEEOT	NVIDIA BlueField-2 P-Series DPU 25GbE Dual-Port SFP56, PCIe Gen3/4 x8, Crypto Enabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHL
MBF2H332A-AENOT	NVIDIA BlueField-2 P-Series DPU 25GbE Dual-Port SFP56, PCIe Gen3/4 x8, Crypto Disabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHL
MBF2H516A-CEEOT	NVIDIA BlueField-2 P-Series DPU 100GbE Dual-Port QSFP56, PCIe Gen4 x16, Crypto Enabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
MBF2H516A-CENOT	NVIDIA BlueField-2 P-Series DPU 100GbE Dual-Port QSFP56, PCIe Gen4 x16, Crypto Disabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
MBF2H516A-EEEOT	NVIDIA BlueField-2 P-Series DPU 100GbE/EDR/HDR100 VPI Dual-Port QSFP56, PCIe Gen4 x16, Crypto Enabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
MBF2H516A-EENOT	NVIDIA BlueField-2 P-Series DPU 100GbE/EDR VPI Dual-Port QSFP56; PCIe Gen4 x16; Crypto Disabled; 16GB on-board DDR; 1GbE OOB management; FHHL
MBF2H516B-CENOT	NVIDIA BlueField-2 P-Series BF2500 DPU Controller, 100GbE Dual-Port QSFP56, PCIe Gen4 x16, Crypto Disabled, 16GB on-board DDR, 1GbE OOB Management, Tall Bracket, FHHL

Model Number	Description
MBF2H516B-EENOT	NVIDIA BlueField-2 P-Series BF2500 DPU Controller, 100GbE/EDR/HDR100 VPI Dual-Port QSFP56, PCIe Gen4 x16, Crypto Disabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
MBF2M322A-AEEOT	NVIDIA BlueField-2 E-Series DPU 25GbE Dual-Port SFP56, PCIe Gen3/4 x8, Crypto, 8GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHL
MBF2M322A-AENOT	NVIDIA BlueField-2 E-Series DPU 25GbE Dual-Port SFP56, PCIe Gen3/4 x8, Crypto Disabled, 8GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHL
MBF2M332A-AEEOT	NVIDIA BlueField-2 E-Series DPU 25GbE Dual-Port SFP56, PCIe Gen4 x8, Crypto, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHL
MBF2M332A-AENOT	NVIDIA BlueField-2 E-Series DPU 25GbE Dual-Port SFP56, PCIe Gen4 x8, Crypto Disabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHL
MBF2M516A-CEEOT	NVIDIA BlueField-2 E-Series DPU 100GbE Dual-Port QSFP56; PCIe Gen4 x16; Crypto Enabled; 16GB on-board DDR; 1GbE OOB management; FHHL
MBF2M516A-CENOT	NVIDIA BlueField-2 E-Series DPU 100GbE Dual-Port QSFP56, PCIe Gen4 x16, Crypto Disabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
MBF2M516A-EEEOT	NVIDIA BlueField-2 E-Series DPU 100GbE/EDR/HDR100 VPI Dual-Port QSFP56, PCIe Gen4 x16, Crypto Enabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
MBF2M516A-EENOT	NVIDIA BlueField-2 E-Series DPU 100GbE/EDR/HDR100 VPI Dual-Port QSFP56; PCIe Gen4 x16; Crypto Disabled; 16GB on-board DDR; 1GbE OOB management; FHHL
900-21004-0030-000	NVIDIA BlueField-2 A30X, P1004 SKU 205, Generic, GA100, 24GB HBM2e, PCIe Passive Dual Slot 230W Gen 4.0, DPU Crypto ON W/ Bkt, 1 Dongle, Black, HF, VCPD
900-21004-0010-000	NVIDIA BlueField-2 A100X, P1004 SKU 230, Generic, GA100, 80GB HBM2e, PCIe Passive Dual Slot 300W Gen 4.0, DPU Crypto ON W/ Bkt, 1 Dongle, Black, HF, VCPD

1.2. Hardware Prerequisites

This quick start guide assumes that an NVIDIA® BlueField® DPU has been installed in a server according to the instructions detailed in your DPU's hardware user guide.

1.3. DOCA Packages

Device	Component	Version	Description
Host	DOCA SDK	0.3.0	Software development kit package for developing host software

Device	Component	Version	Description
	DOCA Runtime	1.2.1	Runtime libraries required to run DOCA-based software applications on host
	DOCA tools	1.2.1	DOCA tools for developers and administrators on host
	Arm emulated (Qemu) development container	3.8.5	Linux-based BlueField Arm emulated container for developers
Target BlueField-2 DPU (Arm)	BlueField OS	3.8.5	BlueField OS image and firmware
	DOCA SDK	0.3.0	Software development kit packages for developing Arm software
	DOCA runtime	1.2.1	Runtime libraries requied to run DOCA-based software applications on Arm
	DOCA tools	1.2.1	DOCA tools for developers and administrators for Arm target

1.4. Supported Operating System

The operating system supported on the BlueField DPU is Ubuntu 20.04.

The following operating systems are supported on the host machine:

- CentOS/RHEL 7.6/8.0/8.2
- Ubuntu 18.04/20.04

1.5. Supported Kernel Versions

Note: Only the following generic kernel versions are supported for DOCA local repo package for host installation (whether by SDKM or manually).

Host Operation System	Kernel Support
CentOS 7.6	3.10.0-957.el7.x86_64
CentOS 8.0	4.18.0-80.el8.x86_64
CentOS 8.2	4.18.0-193.el8.x86_64

Host Operation System	Kernel Support
RHEL 7.6	3.10.0-957.el7.x86_64
RHEL 8.0	4.18.0-80.el8.x86_64
RHEL 8.2	4.18.0-193.el8.x86_64
Ubuntu 18.04	4.15.0-20-generic
Ubuntu 20.04	5.4.0-26-generic

Chapter 2. SDK Manager

NVIDIA SDK Manager supports DOCA installation, including software packages on the host and the BlueField-2 target.

- To use the SDK Manager GUI, please refer to NVIDIA SDK Manager GUI installation guide for DOCA for detailed instructions.
- To use the SDK Manager CLI, please refer to NVIDIA SDK Manager CLI installation guide for DOCA for detailed instructions.

Note: SDK manager installation requires internet connection through out-of-band (OOB) port.

Chapter 3. Manual BlueField Image Installation

This guide provides the minimal first-step instructions for setting up DOCA on a standard system.

3.1. Installation Files

Device	Component	OS	Link
Host	These files contain the following components suitable for their respective OS version. ▶ DOCA SDK v0.3.0 ▶ DOCA Runtime v1.2.1 ▶ DOCA Tools v1.2.1	CentOS/RHEL 7.6	doca-host-repo-rhel76-1.2.1-0.1.5.1.2.006.5.5.2.1.7.0.x86_
		CentOS/RHEL 8.0	doca-host-repo-rhel80-1.2.1-0.1.5.1.2.006.5.5.2.1.7.0.x86_
		CentOS/RHEL 8.2	doca-host-repo-rhel82-1.2.1-0.1.5.1.2.006.5.5.2.1.7.0.x86_
		Ubuntu 18.04	doca-host-repo-ubuntu1804_1.2.1-0.1.5.1.2.006.5.5.2.1.7.0
		Ubuntu 20.04	doca-host-repo-ubuntu2004_1.2.1-0.1.5.1.2.006.5.5.2.1.7.0
	Arm Emulated Development Container	Arm container v3.8.5	bfb_builder_doca_ubuntu_20.04-inbox-5.5.tar
Target BlueField-2 DPU (Arm)	BlueField OS image v3.8.5	Ubuntu 20.04	doca_v1.2.1_bluefield_os_ubuntu_20.04-5 bluefield-5.5-2.1.7.0-3.8.5.12027-1.signed aarch64.bfb
	DOCA SDK v0.3.0		doca-repo-aarch64-ubuntu2004-local_1.2.006-1.5.5.2.1.7.0.bf.3.8.5.12027_
	DOCA Runtime v1.2.1		
	DOCA Tools v1.2.1		

3.2. Software Prerequisites

1. If you wish to continue without the DOCA local repo package for host, install the minimal tools needed on the host to allow managing and flashing new firmware on the BlueField.

   For Ubuntu/Debian

a). Download the DOCA Tools package from Installation Files section for the host.

b). Unpack the deb repo. Run:

```
sudo dpkg -i doca-host-repo-ubuntu<version>_amd64.deb
```

c). Perform apt update. Run:

```
sudo apt-get update
```

d). Run apt install for DOCA SDK, DOCA runtime, DOCA tools.

```
sudo apt install doca-tools
```

For CentOS/RHEL

a). Download the DOCA Tools package from Installation Files section for the x86 host.

b). Unpack the RPM repo. Run:

```
sudo rpm -Uvh doca-host-repo-rhel<version>.x86_64.rpm
```

c). Run `yum install` for DOCA SDK, DOCA runtime, DOCA tools.

```
sudo yum install doca-tools
```

Note: Skip the following step to proceed without the DOCA local repo package for host.

2. Alternatively, to continue with the DOCA local repo package for host installation:

Installing DOCA Local Repo Package on Ubuntu Host

a). Download the DOCA SDK, DOCA Runtime, and DOCA Tools package from Installation Files section for the host.

b). Unpack the deb repo. Run:

```
sudo dpkg -i doca-host-repo-ubuntu<version>_amd64.deb
```

c). Perform apt update. Run:

```
sudo apt-get update
```

d). Run apt install for DOCA SDK, DOCA runtime, DOCA tools.

```
sudo apt install doca-sdk
sudo apt install doca-runtime
sudo apt install doca-tools
```

Installing DOCA Local Repo Package on CentOS Host

a). Download the DOCA SDK, DOCA Runtime, and DOCA Tools package from Installation Files section for the x86 host.

b). Install the following software dependencies. Run:

```
sudo yum install -y epel-release
sudo yum install -y uriparser-devel
sudo yum install -y 'dnf-command(config-manager)'
sudo dnf -y install dnf-plugins-core
sudo yum install -y epel-release
sudo dnf config-manager --set-enabled PowerTools
sudo yum install meson
```

c). Unpack the RPM repo. Run:

```
sudo rpm -Uvh doca-host-repo-rhel<version>.x86_64.rpm
```

d). Run `yum install` for DOCA SDK, DOCA runtime, DOCA tools.

```
sudo yum install doca-sdk
sudo yum install doca-runtime
sudo yum install doca-tools
```

Installing DOCA Local Repo Package on RHEL Host

a). Open a RedHat account.

i. Log into RedHat website via the developers tab.

ii. Create a developer user.

b). Run:

```
subscription-manager register --username=<username> --password=PASSWORD
```

To extract pool ID:

```
subscription-manager list --available --all
...
Subscription Name:   Red Hat Developer Subscription for Individuals
Provides:            Red Hat Developer Tools (for RHEL Server for ARM)
                     ...
                     Red Hat CodeReady Linux Builder for x86_64
...
Pool ID:             <pool-id>
...
```

And use the pool ID for the `Subscription Name` and `Provides` that include `Red Hat CodeReady Linux Builder for x86_64`.

c). Run:

```
subscription-manager attach --pool=<pool-id>
subscription-manager repos --enable codeready-builder-for-rhel-8-x86_64-rpms
yum makecache
```

d). Install the DOCA local repo package for host. Run:

```
rpm -Uvh doca-host-repo-rhel<version>.x86_64.rpm
sudo yum install doca-runtime
sudo yum install doca-sdk
sudo yum install doca-tools
```

e). Sign out from your RHEL account. Run:

```
subscription-manager remove --all
subscription-manager unregister
```

The upgrade takes effect only after mlxfwreset which is performed in later steps.

3. Initialize MST. Run:

```
sudo mst start
```

4. Reset the `nvconfig` params to their default values:

```
sudo mlxconfig -d /dev/mst/mt41686_pciconf0 -y reset

Reset configuration for device /dev/mst/<device>? (y/n) [n] : y
Applying... Done!
-I- Please reboot machine to load new configurations.
```

5. Skip this step if your BlueField DPU is Ethernet only. Please refer to Supported Platforms to learn your DPU type.

   If you have a VPI DPU, the default link type of the ports will be configured to IB. To verify your link type, run:

```
sudo mst start
sudo mlxconfig -d /dev/mst/mt41686_pciconf0 -e q | grep -i link_type
Configurations:                                 Default         Current         Next
 Boot
*        LINK_TYPE_P1                           IB(1)           ETH(2)
 IB(1)
```

```
*         LINK_TYPE_P2                              IB(1)           ETH(2)
 IB(1)
```

Note: If your DPU is Ethernet capable only, then the `sudo mlxconfig -d <device>` command will not provide an output.

If the current link type is set to IB, run the following command to change it to Ethernet:

```
sudo mlxconfig -d /dev/mst/mt41686_pciconf0 s LINK_TYPE_P1=2 LINK_TYPE_P2=2
```

6. Assign a dynamic IP to `tmfifo_net0` interface (RShim host interface).

```
ifconfig tmfifo_net0 192.168.100.1 netmask 255.255.255.252 up
```

7. Verify that RShim is active.

```
sudo systemctl status rshim
```

This command is expected to display `active (running)`. If RShim service does not launch automatically, run:

```
sudo systemctl enable rshim
sudo systemctl start rshim
```

3.3. Image Installation

Users have two options for installing DOCA on the DPU:

- Upgrading the full DOCA image on the DPU (recommended) - this option overwrites the entire boot partition.
- Upgrading DOCA local repo package on the DPU – this option upgrades DOCA components without overwriting the boot partition. Use this option to preserve configurations or files on the DPU itself.

3.3.1. Installing Full DOCA Image on DPU

Note: This step overwrites the entire boot partition.

Ubuntu users are required to provide a unique password that will be applied at the end of the BlueField OS image installation. This password needs to be defined in a `bf.cfg` configuration file.

To set the password for the "ubuntu" user:

1. Create password hash. Run:

```
# openssl passwd -1
Password:
Verifying - Password:
$1$3B0RIrfX$TlHry93NFUJzg3Nya00rE1
```

2. Add the password hash in quotes to the `bf.cfg` file:

```
# sudo vim bf.cfg
ubuntu_PASSWORD='$1$3B0RIrfX$TlHry93NFUJzg3Nya00rE1'
```

When running the installation command, use the `--config` flag to provide the file containing the password:

```
sudo bfb-install --rshim <rshimN> --bfb <image_path.bfb> --config bf.cfg
```

Note: If `--config` is not used, then upon first login to the BlueField device, users will be asked to update their password.

The following is an example of Ubuntu installation assuming the "pv" Linux tool has been installed (to view the installation progress).

```
sudo bfb-install --rshim rshim0 --bfb DOCA_<version>-aarch64.bfb --config bf.cfg
Pushing bfb
1.08GiB 0:00:57 [19.5MiB/s] [      <=>    ]
Collecting BlueField booting status. Press Ctrl+C to stop…
INFO[BL2]: start
INFO[BL2]: DDR POST passed
INFO[BL2]: UEFI loaded
INFO[BL31]: start
INFO[BL31]: runtime
INFO[UEFI]: eMMC init
INFO[UEFI]: eMMC probed
INFO[UEFI]: PCIe enum start
INFO[UEFI]: PCIe enum end
INFO[MISC]: Ubuntu installation started
INFO[MISC]: Installation finished
INFO[MISC]: Rebooting...
```

Note: This installation sets up the OVS bridge.

3.3.2. Installing DOCA Local Repo Package on DPU

Note: If you have already installed BlueField OS image, be aware that the DOCA SDK Runtime tools are already contained in the BFB, and that this installation is not mandatory.

Note: Before installing DOCA on the target DPU, make sure the out-of-band interface (mgmt) is connected to the internet.

1. Download the DOCA SDK, DOCA Runtime, and DOCA Tools package from section Installation Files.
2. Copy deb repo package into BlueField. Run:
   ```
   sudo scp -r doca-repo-aarch64-ubuntu2004-local_<version>_arm64.deb
    ubuntu@192.168.100.2:/tmp/
   ```
3. Unpack the deb repo. Run:
   ```
   sudo dpkg -i doca-repo-aarch64-ubuntu2004-local_<version>_arm64.deb
   ```
4. Run apt update.
   ```
   sudo apt-get update
   ```
5. Run apt install for DOCA SDK, DOCA runtime, DOCA tools:
   ```
   sudo apt install doca-sdk
   sudo apt install doca-runtime
   sudo apt install doca-tools
   ```

3.4. Firmware Upgrade

To upgrade firmware:

1. SSH to your BlueField device via 192.168.100.2 (preconfigured). The default credentials for Ubuntu are as follows:
   - Username: ubuntu
   - Password: unique password

   For example:
   ```
   ssh ubuntu@192.168.100.2 Password: <configured-password>
   ```
2. Upgrade firmware in BlueField DPU. Run:
   ```
   sudo /opt/mellanox/mlnx-fw-updater/mlnx_fw_updater.pl
   ```
   Example output:
   ```
   Device #1:
   ----------

     Device Type:      BlueField-2
     [...]
     Versions:         Current        Available
        FW             <Old_FW>       <New_FW>
   ```
3. For the firmware upgrade to take effect:

   a). Run the following command on the BlueField DPU and host:
   ```
   sudo mst start
   ```
   b). Run the command below on the BlueField DPU and immediately afterwards on the host. *Do not* wait for the command to complete on the BlueField DPU before issuing the command on the host.
   ```
   sudo mlxfwreset -d /dev/mst/<device> -l 3 -y reset
   ```

   Note: If your BlueField device is a controller or if you are performing remote install, you must power cycle the BlueField.

3.5. Post-installation Procedure

1. Restart the driver. Run:
   ```
   host$ sudo /etc/init.d/openibd restart
   Unloading HCA driver:                                      [  OK  ]
   Loading HCA driver and Access Layer:                       [  OK  ]
   ```
2. Configure the physical function (PF) interfaces.
   ```
   host$ sudo ifconfig <interface-1> <network-1/mask> up
   host$ sudo ifconfig <interface-2> <network-2/mask> up
   ```
   For example:
   ```
   host$ sudo ifconfig p2p1 192.168.200.32/24 up
   host$ sudo ifconfig p2p2 192.168.201.32/24 up
   ```
   Pings between the source and destination should now be operational.

Chapter 4. Setting Up Build Environment for Developers

For full instructions about setting up a development environment, refer to the NVIDIA DOCA Developer Guide.

Chapter 5. Installing CUDA on NVIDIA Converged Accelerator

NVIDIA® CUDA® is a parallel computing platform and programming model developed by NVIDIA for general computing GPUs.

This section details the necessary steps to set up CUDA on your environment. This section assumes that a BFB image has already been installed on your environment.

To install CUDA on your converged accelerator:

1. Download and install the latest NVIDIA Data Center GPU driver.
2. Download and install CUDA.

Note: It is important to select a compatible NVIDIA Data Center GPU driver and CUDA version. In the procedure below, NVIDIA driver 495 with CUDA11-5 are used as an example, however the same steps can be done with NVIDIA driver 470 and CUDA-11.4. For more information about CUDA and driver compatibility please refer to "NVIDIA CUDA Toolkit Release Notes".

5.1. Downloading and Installing NVIDIA Data Center GPU Driver

NVIDIA Data Center GPU driver installation is done using a run file. First, you must download the run file of the relevant driver version. This link contains the run file which installs the driver, version 495, based on your Linux distribution and your system architecture (should be Arm64).

This section details the necessary steps to set up CUDA on your environment. It assumes that a BFB image has already been installed on your environment.

The downloaded run file should be executed from the converged accelerator. There are multiple ways of doing that, however the easiest is downloading it locally and copying it to the converged accelerator.

Execute the run file:

```
./<runfile-name>
```

Test that the driver installation completed successfully. Run:

```
nvidia-smi
```

Output example:

```
Thu Oct 28 11:28:13 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 495.08       Driver Version: 495.08       CUDA Version: 11.5     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               Mig M. |
|===============================+======================+======================|
|   0  NVIDIA BF A10       Off  | 00000000:06:00.0 Off |                    0 |
|  0%   34C    P0    83W / 225W |      0MiB / 22731MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
| No running process found                                                    |
+-----------------------------------------------------------------------------+
```

5.2. Downloading and Installing CUDA

Downloading and installing CUDA can be done in the CUDA Toolkit 11.5 Downloads webpage.

Note: Select the Linux distribution and version relevant for your environment and make sure the installer type is "runfile (local)".

You may encounter issues when trying to run the `wget` commands found in the above link on the converged accelerator. In that case, follow these steps:

1. Download the `*.pin` file according to your Linux distribution and version and copy it to the converged accelerator.

   This link downloads the `*.pin` file for Ubuntu 20.04 for Arm64 (`sbsa`) system architecture.
2. Download the CUDA repo.

   Select the relevant CUDA repo according to the Linux distribution and version on your environment. This link contains the CUDA-11.5 Toolkit for Ubuntu 20.04 for Arm64 system architecture (`*arm64.deb`).
3. Install the CUDA packages.

   Run the following commands on the converged card:

   ```
   sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-repository-pin-600
   sudo dpkg -i cuda-repo-ubuntu2004-11-5-local_11.5.0-495.29.05-1_arm64.deb
   sudo apt-key add /var/cuda-repo-ubuntu2004-11-5-local/7fa2af80.pu
   sudo apt-get update
   sudo apt-get -y install cuda-toolkit-11-5
   ```
4. Run `vectorAdd` sample (located under `/usr/local/cuda11-5/samples/0_Sample/vectorAdd`) to verify that the installation completed successfully.

   ```
   ./vectorAdd
   ```

If the `vectorAdd` sample worked as expected, it should output "`Test Passed`".

Note: When running `vectorAdd`, if it seems that the GPU is slow or stuck, stop execution and run the following command:

```
sudo setpci -v -d ::0302 800.L=201 # CPL_VC0 = 32
```

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 nor any of its direct or indirect subsidiaries and affiliates (collectively: "NVIDIA") make no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assume no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

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

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

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

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

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

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

Trademarks

NVIDIA, the NVIDIA logo, and Mellanox are trademarks and/or registered trademarks of Mellanox Technologies Ltd. and/or NVIDIA Corporation in the U.S. and in other countries. The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a world¬wide basis. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com