NVIDIA DOCA

Installation Guide

MLNX-15-060470 _v1.1 | August 2021

Table of Contents

Chapter 1. Introduction

There are two ways 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	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	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	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	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	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	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	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	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
MBF2H516B-EENOT	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	BlueField-2 E-Series DPU 25GbE Dual-Port SFP56, PCIe Gen3/4 x8, Crypto, 8GB on-board DDR, 1GbE OOB management, Tall Bracket, HHHL

Model Number	Description
MBF2M322A-AENOT	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	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	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	BlueField-2 E-Series DPU 100GbE Dual-Port QSFP56; PCIe Gen4 x16; Crypto Enabled; 16GB on-board DDR; 1GbE OOB management; FHHL
MBF2M516A-CENOT	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	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	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

1.2. Hardware Prerequisites

This quick start guide assumes that an NVIDIA® BlueField®-2 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 (metapackages)	DOCA SDK	0.2	Software development kit package for developing host software
	DOCA Runtime	1.1	Runtime libraries required to run DOCA-based software applications on host
	DOCA tools	1.1	DOCA tools for developers and administrators on host
	Arm emulated (Qemu) development container	3.7	Linux-based BlueField Arm emulated container for developers
Target BlueField-2 DPU (Arm)	BlueField OS	3.7	BlueField OS image and firmware
	DOCA SDK	0.2	Software development kit packages for

Device	Component	Version	Description
			developing Arm software
	DOCA runtime	1.1	Runtime libraries requied to run DOCA-based software applications on Arm
	DOCA tools	1.1	DOCA tools for developers and administrators for Arm target

1.4. Supported Operating System

Installation Method	Host Machine	Target Hardware (BlueField-2 DPU)
Manual installation	CentOS 7.6/8.0/8.2	Ubuntu 20.04
	Ubuntu 18.04/20.04	
SDK Manager installation	CentOS 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 metapackage 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
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.

Developers must have access to DOCA to install the relevant SDK packages. Visit the DOCA developer zone landing page to request access.

Chapter 3. Manual BlueField Image Installation

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

3.1. Installation Files

Device	Component	OS	Link
Host (metapackages)	These files contain the following components suitable for their respective OS version. ▶ DOCA SDK v0.2 ▶ DOCA Runtime v1.1 ▶ DOCA Tools v1.1	CentOS 7.6	doca-repo-rhel76-1.1-0.1.7.1.1.024.5.4.1.0.3.0.x86_64
		CentOS 8.0	doca-repo-rhel80-1.1-0.1.7.1.1.024.5.4.1.0.3.0.x86_64
		CentOS 8.2	doca-repo-rhel82-1.1-0.1.7.1.1.024.5.4.1.0.3.0.x86_64
		Ubuntu 18.04	doca-repo-ubuntu1804_1.1-0.1.7.1.1.024.5.4.1.0.3.0_a
		Ubuntu 20.04	doca-repo-ubuntu2004_1.1-0.1.7.1.1.024.5.4.1.0.3.0_a
	Arm Emulated Development Container	Arm container v3.7	bfb_builder_doca_ubuntu_20.04-inbox-5.4.tar
Target BlueField-2 DPU (Arm)	MFT v3.7	CentOS 7.6	mft-repo-centos7.6-4.17.0-106.x86_64.rpm
		CentOS 8.x	mft-repo-centos8.x-4.17.0-106.x86_64.rpm
		Ubuntu 18.04	mft-repo-amd64-ubuntu2004-local_4.17.0-106_amd64.deb
		Ubuntu 20.04	
	RShim v3.7	CentOS 7.6	rshim-2.0.6-1.ga97dc5d.el7.centos.x86_64
		CentOS 8.x	
		Ubuntu 18.04	rshim_2.0.6-1.ga97dc5d_amd64.deb
		Ubuntu 20.04	

Device	Component	OS	Link
	BlueField OS image v3.7	Ubuntu 20.04	DOCA_v1.1_BlueField_OS_Ubuntu_20.04-bluefield-5.4-1.0.3.0-3.7.0.11805-1-aarch64.bfb
	DOCA SDK v0.2		doca-repo-aarch64-ubuntu2004-local_1.1-0.5.4.1.0.3.0.bf.3.7.0.11805_arm
	DOCA Runtime v1.1		
	DOCA Tools v1.1		

3.2. Software Prerequisites

1. Download the following packages listed in the table under section Installation Files depending on the OS of the host you are using:
   - BlueField OS image
   - MFT
   - DOCA metapackages for host

   Note: Alternatively, you may choose not to install the DOCA metapackage for the host (step 2). In this case, you must download the RShim package separately:
   - MFT
   - RShim
2. To continue with the DOCA metapackage installation:

   a). Uninstall the driver if it is already installed. Run:
   ```
   ofed_uninstall.sh
   ```
   b). Install MFT.
   - For Ubuntu

     Uninstall the previous MFT package if it exists. Run:
     ```
     sudo apt remove --purge mft  mft-oem mft-pcap mft-repo-amd64-ubuntu2004-local mft kernel-mft-dkms -y
     sudo apt-get autoremove
     ```
     Then reinstall MFT. Run:
     ```
     sudo dpkg -i mft-repo-amd64-ubuntu2004-local_<version>_amd64.deb
     apt-get update
     apt install mft-meta
     ```
   - For CentOS, run:
     ```
     rpm -Uvh mft-repo-4.16.3-12.x86_64.rpm
     yum install mft-meta
     ```

   c). Perform the instructions in the following DOCA installation on host sections.

   Installing DOCA Metapackages 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:

```
dpkg -i <repo_file>
```

For example:

```
sudo dpkg -i doca-repo-ubuntu2004_1.1-0.0.5.5.4.0.6.3.0_amd64.deb
```

c). Perform apt update. Run:

```
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 Metapackages 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 meta package dependencies. Run:

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

c). Unpack the deb repo. Run:

```
rpm -Uvh <repo_file>.rpm
```

For example:

```
rpm -Uvh doca-repo-rhel82-1.1-0.0.5.5.4.0.6.3.0.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
```

As part of the metapackage installation the relevant firmware version is installed. The script print-out will display the Current and Available firmware versions as shown in the following example:

```
Device #1:
----------

  Device Type:      BlueField-2
  [...]
  Versions:         Current        Available
     FW             <Old_FW>       <New_FW>
```

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

3. To continue without the DOCA metapackages:

a). Install RShim.

- For Ubuntu, run:

```
sudo dpkg --force-all -i rshim-<version>.deb
sudo dpkg --force-all -i mft-<version>.deb
```

- For CentOS, run:

```
sudo rpm -Uhv rshim-<version>.rpm
sudo rpm -Uhv mft-<version>.rpm
```

b). Install MFT.

- For Ubuntu

Uninstall the previous MFT package if it exists. Run:

```
sudo apt remove --purge mft  mft-oem mft-pcap mft-repo-amd64-ubuntu2004-
local mft kernel-mft-dkms -y
sudo apt-get autoremove
```

Then reinstall MFT. Run:

```
sudo dpkg -i mft-repo-amd64-ubuntu2004-local_<version>_amd64.deb
apt-get update
apt install mft-meta
```

- For CentOS, run:

```
rpm -Uvh mft-repo-4.16.3-12.x86_64.rpm
yum install mft-meta
```

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

```
sudo mlxconfig -d /dev/mst/<device> -y reset

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

> **Note:** You may retrieve the `<device>` parameter by using the command `sudo mst status -v`.

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/<device> -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 <device> s LINK_TYPE_P1=2 LINK_TYPE_P2=2
```

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

   - For Ubuntu, edit the file `/etc/netplan/01-netcfg.yaml` by adding the following lines:

```
tmfifo_net0:
  addresses: [192.168.100.1/24]
  dhcp4: false
```

   Example:

```
sudo cat /etc/netplan/01-netcfg.yaml
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    eno1:
```

```
      dhcp4: yes
    tmfifo_net0:
      addresses: [192.168.100.1/24]
      dhcp4: no
```

- For CentOS, create the file `/etc/sysconfig/network-scripts/ifcfg-tmfifo_net0` and set the following lines:

```
DEVICE=tmfifo_net0
BOOTPROTO=none
ONBOOT=yes
PREFIX=24
IPADDR=192.168.100.1
NM_CONTROLLED=no
```

7. Execute network restart for implemented `tmfifo_net0` static configuration.

   - For CentOS:

   ```
   /etc/init.d/network restart
   ```

   - For Ubuntu:

   ```
   /etc/init.d/networking restart
   ```

8. 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

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
   ```

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

   ```
   sudo bfb-install --rshim rshim0 --bfb
    DOCA_v1.0_BlueField_OS_Ubuntu_20.04-5.3-1.0.0.0-3.6.0.11699-1-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: The `--config` flag is necessary for Ubuntu users only. If this flag is not used by Ubuntu users, then upon first login to the BlueField device, they will be asked to update their password.

Note: This installation sets up the OVS bridge.

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: <unique-password>
   ```
2. Perform the following step only if:
   - Your BlueField device is a controller
   - If you did not install the DOCA metapackages

   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 you must power cycle the controller as `mlxfwreset` is not supported.

3.5. Post-installation Procedure

1. Restart the driver. Run:

   ```
   sudo /etc/init.d/openibd restart
   Unloading HCA driver:                                      [  OK  ]
   Loading HCA driver and Access Layer:                       [  OK  ]
   ```

2. Configure the physical function (PF) interfaces.

   ```
   sudo ifconfig <interface-1> <network-1/mask> up
   sudo ifconfig <interface-2> <network-2/mask> up
   ```

   For example:

   ```
   sudo ifconfig p2p1 192.168.200.32/24 up
   sudo ifconfig p2p2 192.168.201.32/24 up
   ```

   Pings between the source and destination should now be operational.

Chapter 4. Installing DOCA on BlueField DPU

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:
   ```
   scp -r doca-repo-aarch64-ubuntu2004-
   local_1.0-0.5.3.0.3.6.bf.3.6.0.11691_arm64.deb ubuntu@192.168.100.2:/tmp/
   ```
3. Unpack the deb repo. Run:
   ```
   dpkg -i <repo_file>
   ```
   For example:
   ```
   sudo dpkg -i doca-repo-aarch64-ubuntu2004-
   local_1.0-0.5.3.0.3.6.bf.3.6.0.11691_arm64.deb
   ```
4. Run apt update.
   ```
   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
   ```

Chapter 5. Setting Up Build Environment Container for Developers

1. Make sure Docker is installed on your host. Run:
   ```
   docker version
   ```
   If docker is not installed, please visit the official Install Docker Engine for installation instructions.
2. Install QEMU on the host.

   Note: This step is for hosts only. If you are working on an aarch64 host, please move to the next step.

   - For Ubuntu host, run:
     ```
     sudo apt-get install qemu binfmt-support qemu-user-static
     sudo docker run --rm --privileged multiarch/qemu-user-static --reset -p yes
     ```
   - For CentOS 7.x host, run:
     ```
     sudo yum install epel-release
     sudo yum install qemu-system-arm
     ```
   - For CentOS 8.0 or 8.2 host, run:
     ```
     sudo yum install epel-release
     sudo yum install qemu-kvm
     ```
   - For Fedora host, run:
     ```
     sudo yum install qemu-system-aarch64
     ```
3. If you are using CentOS or Fedora on the host, verify if `qemu-aarch64.conf` exists. Run:
   ```
   $ cat /etc/binfmt.d/qemu-aarch64.conf
   ```
   If it does not, run:
   ```
   echo ":qemu-aarch64:M::\x7fELF
   \x02\x01\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x02\x00\xb7:\xff\xff\xff\xff\xff
   \xff\xff\xfc\xff\xff\xff\xff\xff\xff\xff\xff\xfe\xff\xff:/usr/bin/qemu-aarch64-
   static:" > /etc/binfmt.d/qemu-aarch64.conf
   ```
4. If you are using CentOS or Fedora on the host, restart system binfmt. Run:
   ```
   $ sudo systemctl restart systemd-binfmt
   ```
5. Load the docker image.

   a). Make sure the docker service is started. Run:
   ```
   systemctl daemon-reload
   systemctl start docker
   ```

b). Go to the location the tar file is saved at and run the following command from the host:

```
sudo docker load -i <filename>
```

For example:

```
sudo docker load -i bfb_builder_ubuntu20.04-5.3-1.0.0.0-3.6.0.11699-1.tar
```

Note: The loading process may take a while. After the image is loaded, you can find its ID using the command `docker images`.

6. Run the docker image.

```
sudo docker run -v <source-code-folder>:<dest-folder-on-docker> --privileged -it
 -e container=docker <image-name/ID>
```

For example, if the source code folder is `/<...>/buildEnv`, the destination folder on the docker is `/app`, and the image is the one downloaded in the previous step, the command will look like this:

```
sudo docker run -v /<...>/buildEnv:/app --privileged -it -e container=docker
 mellanox/bluefield:bfb_builder_ubuntu20.04-5.3-1.0.0.0-3.6.0.11699-1
```

Or, for example, if you use a loaded image with the ID 185c50ecb31d, the command will be:

```
sudo docker run -v /<...>/buildEnv:/app --privileged -it -e container=docker
 185c50ecb31d
```

After you run this command, you will have a shell inside the container, where you can build your project using the `gcc` command.

Note: Please make sure you map a folder that everyone has write privileges to. Otherwise, the docker will not be able to write the output file to it.

Note: The folder will be mapped to the "dest" folder. In this example the folder `/app` inside the docker will be mapped to `/<...>/buildEnv`.

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. 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