



## **DPL Debugger**

# Table of contents

Introduction

---

Executing the DPL Debugger

---

Using the DPL Debugger

---

Open Debug Session File

---

Packets List

---

Filter

---

Parser Graph

---

Debug Points

---

Variables

---

Source Code

---

Log

---

System Info

---

Packet Dissector

---

Hex Dump

---

Options

---

This page describes the DPL Debugger tools, which are used to debug pipeline packets processed by DPL programs on the NVIDIA® BlueField® DPU.

## Introduction

The DPL Debugger is a GUI application that can run a live debug session or open a pre-recorded debug session file captured using the `dpl_nspect debug` command.

A debug session visualizes the flow of network packets through a DPL program's pipeline, allowing developers to trace each packet's path in detail. For more information, refer to the documentation on the [nv\\_send\\_debug\\_pkt](#) extern function.

A debug session visualizes the flow of network packets through a DPL program's pipeline, allowing developers to trace packet details throughout its course. For more information, refer to the documentation on the [nv\\_send\\_debug\\_pkt](#) extern function.

### Note

The current version of the debugger only displays debug packets received by the DPU (Rx path) or packets arriving from the primary wire port (P0) in the network-to-host direction.

## Executing the DPL Debugger

Running `dpl_debugger .sh` without any arguments launches the GUI application.

To open and debug a pre-recorded session specify the session archive when executing the program:

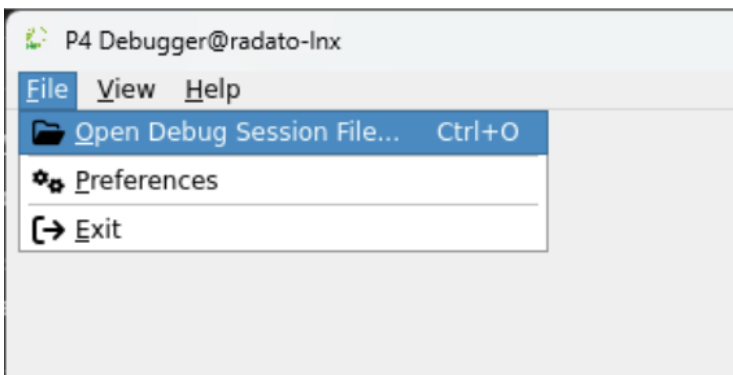
```
dpl_debugger.sh my-debug-archive.tar.gz
```

## Using the DPL Debugger

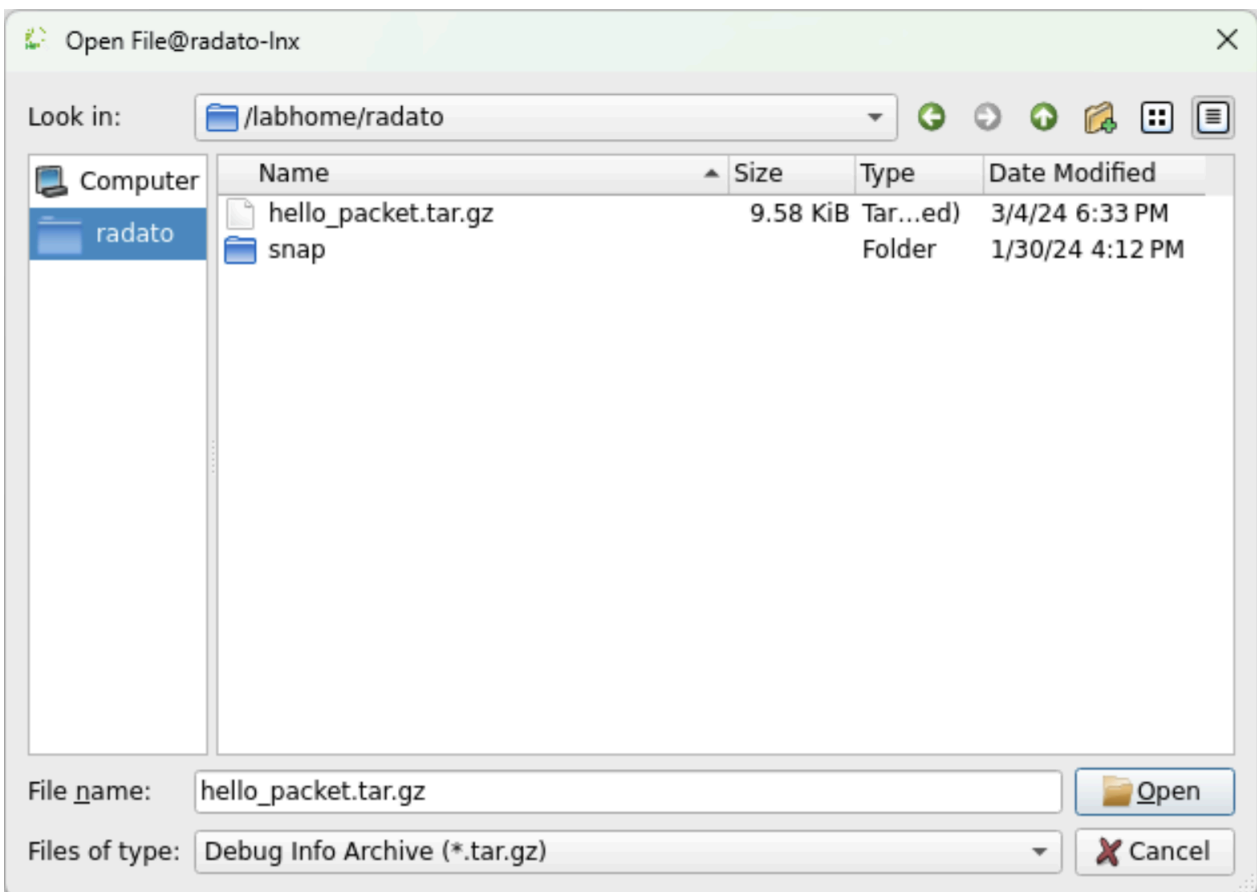
### Open Debug Session File

To open a pre-recorded debug session in the DPL Debugger GUI:

1. Select the **File** → **Open Debug Session File**.



2. Choose the appropriate debug session file (`hello_packet.tar.gz` in this example).



**i** Info

The `tar.gz` archive should be the output of a recorded debug session created using the `dpl_nspect debug` command.

## Packets List

The left pane of the DPL Debugger displays the **Packets** list, which includes the following fields:

- Date Time – The timestamp when the packet entered the DPL pipeline
- Time Since Start – The elapsed time since the first packet entered the DPL pipeline
- Size – The packet size
- Ingress Port – The port through which the packet entered the pipeline
- Protocol – The network protocol associated with the packet

The screenshot displays the DPL Debugger interface with the following components:

- Packets List:** A table showing 16 packets with columns for Date Time, Time Since Start, Size, and Ingress Port. The selected packet (index 109) has a timestamp of 2025-02-02 15:14:54, a size of 380 bytes, and an ingress port of 109.
- Debug Points:** A table with columns for P4 Object, Location, Table[Entry], and Match Key. The selected point is a Table at myprogram.p4:196, matching on headers.vxlan.vni = 0x10 (exact).
- P4 Variables:** A table with Name and Value columns. The selected variable is tenant\_id with a value of 0.
- Dissector:** A table showing the protocol stack: Ethernet, IPv4, and UDP.
- Hexdump:** A table displaying the raw bytes of the packet, starting with a0 88 c2 75 6b 90 00 00 00 00 00 00 00 45 00.
- Pipeline Parser:** A flow graph showing the packet's path through various parsers and tables, including inner\_ethernet, vxlan, and ipsec\_esp.
- Log:** A text area showing the execution of the DPL pipeline, including the nv\_send\_debug\_pkt() function call.

# Filter

The **Packets** list can be filtered using the fields located at the bottom left of the GUI.



To apply a filter:

1. Enter a condition (e.g., "packet ID is not 0").
2. Click the **+** button to add the filter.
3. Click the **Apply** button to apply the filter.

Packets

Date Time	Time Since Start (ms)	Size	Ingress Port	Protocol
2025-01-13 20:55:49	0.000000	109	256	VXLAN
2025-01-13 20:56:07	18408.245433	109	256	VXLAN

Filters

ID	Is Not	0	
ID	Is Not	0	
<input type="button" value="✓ Apply"/>		<input type="button" value="🗑️ Clear All"/>	

Showing 2 out of 2 packets

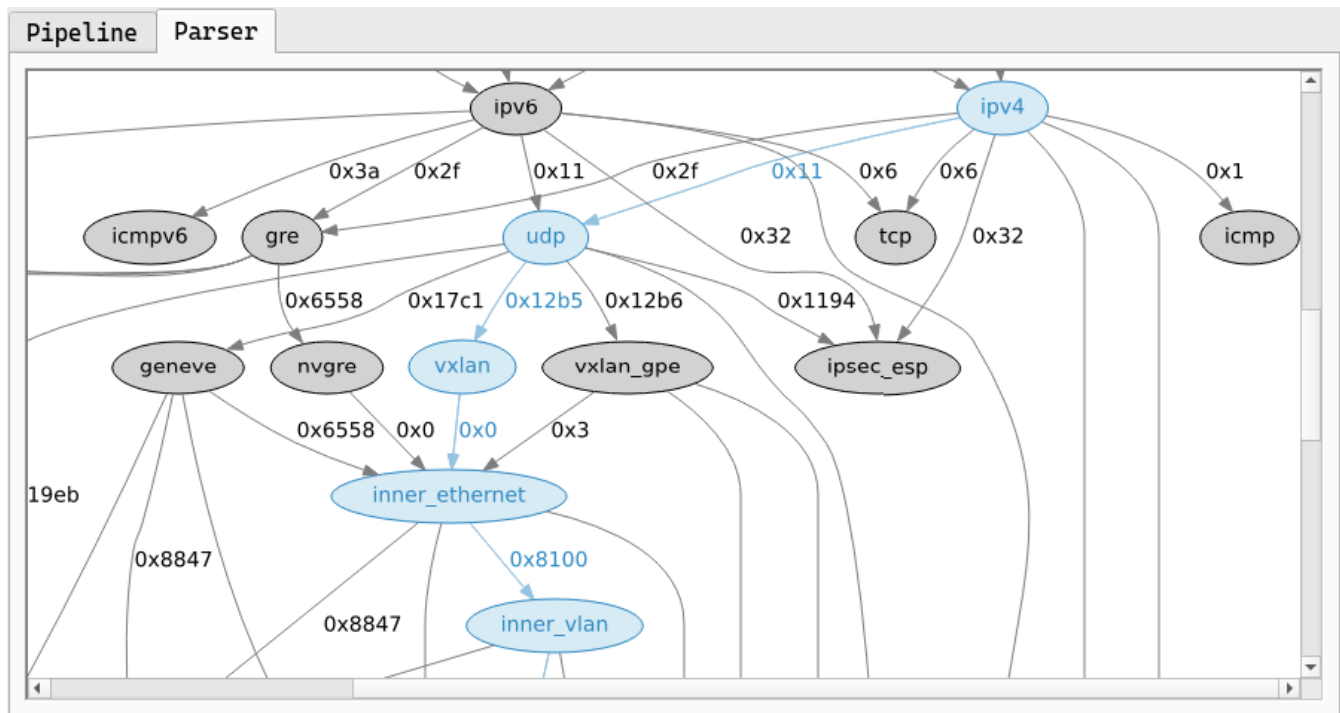
To reset the **Packets** list to its original state:

1. Click the **Clear All** button.
2. Click **Apply**.

# Parser Graph

The **Parser** graph, located at the bottom of the GUI, outlines the possible routes a packet can take within the DPL pipeline.

- Blue nodes ( `inner_ethernet` in the example) indicate the nodes the packet has traversed.
- The graph consists of two types of nodes:
  - Flex nodes – Defined by the DPL programmer
  - Static (fixed) nodes – Built-in components of the hardware



This visualization helps developers understand the packet's path through the parsing stages.

# Debug Points

Selecting a packet from the **Packets** List displays its pipeline stages in the Debug Points view at the top of the GUI:

- Each row corresponds to a specific source location in the DPL program code



- The table cells are interactive—clicking a cell highlights the relevant source location, when applicable

Debug Points			
P4 Object	Location	Table[Entry]	Match Key = Value/Mask (Type)
1	Control	myprogram.p4:225	
2	Table	myprogram.tenant[1]	headers.vxlan.vni = 0x10 (exact)
			tenant_id = 0x1 (exact)
			headers.vlan.vlan_id = 0x0 (exact)

This allows developers to trace a packet’s processing path and correlate it with the corresponding code sections.

## Variables

Selecting a pipeline stage (either a single cell or an entire row) displays variable information in two tabs:

- P4 Variables – Shows DPL variables defined in the program and their values at the selected pipeline stage.
- Raw Variables – Provides a low-level view of internal variables, such as registers, samplers, and other hardware-related data.

P4 Variables		Raw Variables	
Name	Value		
1	table_d_key 0	<ul style="list-style-type: none"> <li>▸ Registers</li> <li>▸ Samplers</li> <li>Controller Meta 0x00000000</li> <li>Parser Bitmap 0x00000000</li> </ul>	
2	session_id 0		
3	tenant_id 0		

This helps developers inspect both high-level program variables and hardware-level details while debugging.

## Source Code

Selecting a cell in a pipeline stage opens the DPL program source code in a new tab on the right-hand side of the DPL Debugger GUI. When applicable, the relevant source location is highlighted.

The debugger locates the DPL source file using:

- Search paths defined in the **Settings**
- Manual file selection, if the source file is not found automatically

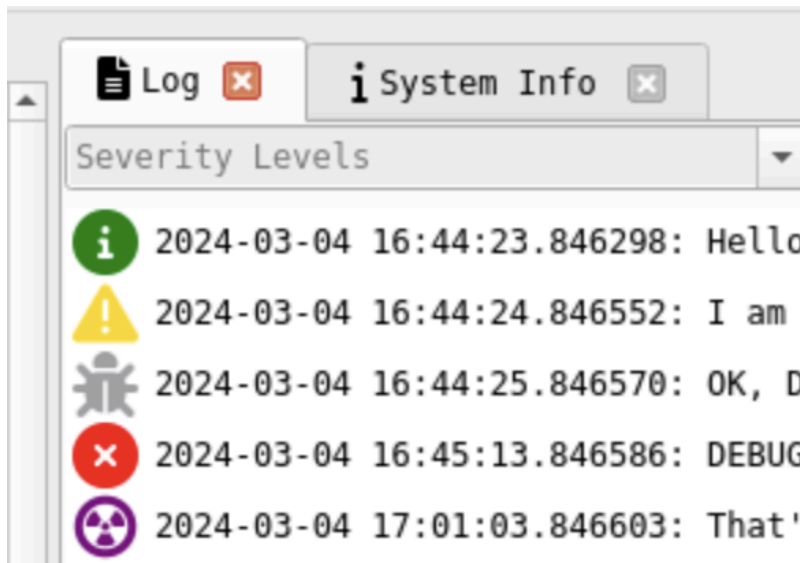
```
Log x i System Info x myprogram.p4 x
214
215     apply {
216         miss_data.session_id = 0;
217         tenant_id = 0;
218         session_id = 0;
219         table_a_key = 0;
220         table_b_key = 0;
221         table_c_key = 0;
222         table_d_key = 0;
223
224         if (headers.vxlan.isValid()) {
225             nv_send_debug_pkt();
226             tenant.apply();
227             if (flow_table.apply().miss) {
228                 route.apply();
229             }
230             session.apply();
231
232             table_a.apply();
233             if (table_b.apply().miss) {
234                 table_a_key = 0xff;
235                 table_a.apply();
236                 if (table_b.apply().miss) {
237                     nv_send_to_controller(miss_d
238                 }
239             }
240             if (table_c.apply().miss) {
241                 table_a_key = 0xffff;
242                 table_a.apply();
243                 if (table_b.apply().miss) {
244                     nv_send_to_controller(miss_d
245                 }
246                 table_c.apply();
247             }
248             table_d.apply();
249         }
250         // On miss the packet goes to the contro
251         nv_send_to_controller(miss_data);
```

## Log

The **Log** tab, located on the right side of the GUI, displays internal logs from the DPL Runtime daemon.

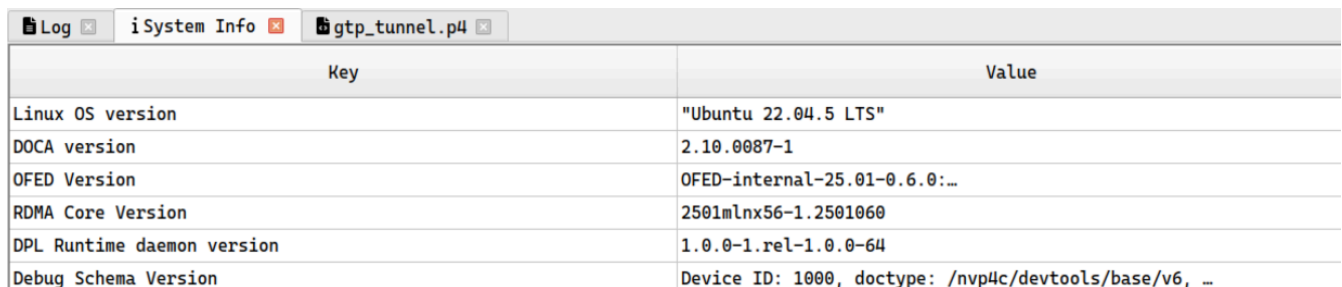
### **Note**

Currently, no log messages are being sent.



## System Info

The **System Info** tab, located on the right side of the GUI, displays details about the system where debugging is performed.



Key	Value
Linux OS version	"Ubuntu 22.04.5 LTS"
DOCA version	2.10.0087-1
OFED Version	OFED-internal-25.01-0.6.0:...
RDMA Core Version	2501mlnx56-1.2501060
DPL Runtime daemon version	1.0.0-1.rel-1.0.0-64
Debug Schema Version	Device ID: 1000, doctype: /nvp4c/devtools/base/v6, ...

## Packet Dissector

The **Dissector** view, located in the center of the GUI, analyzes a packet's structure when a pipeline stage is selected.

Dissector (displaying 109 bytes)

Field	Value
▼ ETHERNET	
dst_addr	a0:88:c2:75:6b:90
src_addr	00:00:00:00:00:00
ether_type	Internet Protocol version 4 (IPv4)
▶ IPV4	
UDP	

## Hex Dump

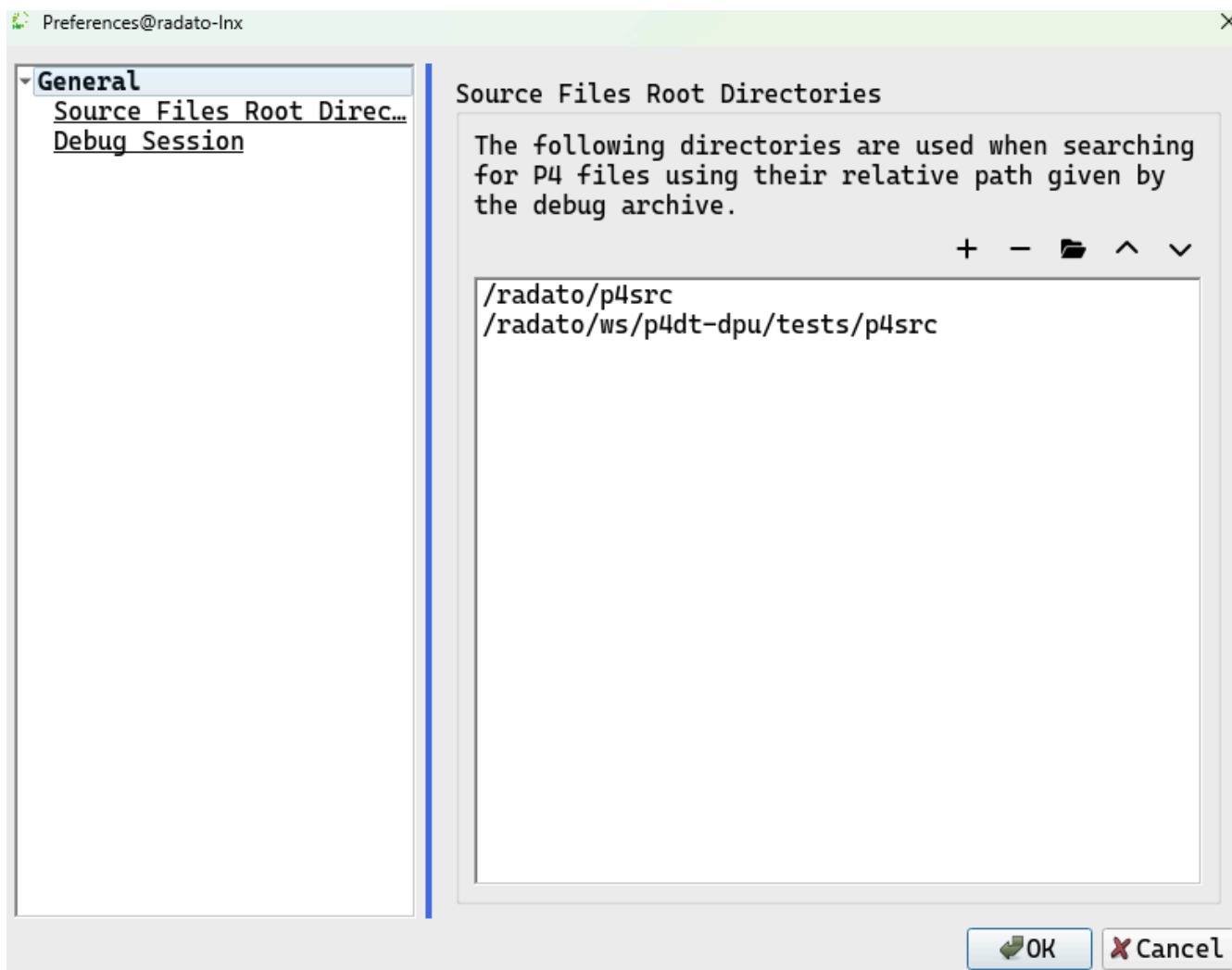
The **Hex Dump** view, located in the center of the GUI, displays a raw hexadecimal representation of the packet when a pipeline stage is selected.

Hexdump (displaying 109 bytes)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a0	88	c2	75	6b	90	00	00	00	00	00	08	00	45	00	
00	5f	00	01	00	00	40	11	7c	8b	7f	00	00	01	7f	00
00	01	12	b5	12	b5	00	4b	00	d7	0c	00	00	03	00	00
10	00	a0	88	c2	75	6b	90	40	b0	76	0d	b6	e6	81	00
00	00	00	00	45	00	00	00	00	01	00	00	40	11	7f	8b

## Options

The **File → Preferences** menu opens the **Preferences** window, where users can configure various debugger and debug session settings.



**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.

**Trademarks**

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

© Copyright 2025, NVIDIA. PDF Generated on 04/24/2025