



## **DOCA Development Best Practices**

# Table of contents

|                     |   |
|---------------------|---|
| Capability Checking | 3 |
| Debuggability       | 6 |

The following sub-sections describe some best practices DOCA SDK users/developers should consider when using DOCA SDK.

- Capability Checking
- Debuggability

---

# Capability Checking

An application that uses a DOCA Device relies on a subset of features for it to function as designed. As such, it is recommended to check whether these features exist for the selected DOCA Device. To achieve this, DOCA SDK exposes capabilities which are a set of APIs with a common look and feel, as described on this page.

The application is expected to use these capability APIs prior to any use of DOCA SDK APIs (Core, libraries) to fail as soon as possible (before initializing any resource) and to be able to implement fallback flows instead of getting error unexpectedly in the application flow.

## Device Capability

An application that uses DOCA Core APIs may need to identify the specific DOCA Device to work based on specific capabilities.

For that, `doca_devinfo` and `doca_devinfo_rep` expose APIs with the prefix `doca_devinfo_cap_*`/`doca_devinfo_rep_cap_*`. For example:

```
doca_error_t
doca_devinfo_cap_is_hotplug_manager_supported(const struct
doca_devinfo *devinfo, uint8_t *is_hotplug_manager);
doca_error_t
doca_devinfo_rep_cap_is_filter_emulated_supported(const struct
doca_devinfo *devinfo, uint8_t *filter_emulated_supported);
```

## Library Capability

Each DOCA library exposes a set of capability APIs for the following purposes:

- Querying the maximum/minimum valid values of a configuration property of the library context or a library task
- Validating whether a library task is supported for a specific DOCA Device

All library capability API starts with the prefix `doca_<library_name>_cap_*`. Moreover:

- Configuration limitation capability APIs start with the prefix `doca_<library_name>_cap_[task_<task_type>]_get_min/max_*`
- Task supported capability APIs have the naming template `doca_<library_name>_cap_task_<task_type>_is_supported`

For example, DOCA DMA exposes:

```
doca_error_t doca_dma_cap_task_memcpy_is_supported(const struct
doca_devinfo *devinfo);
doca_error_t doca_dma_cap_get_max_num_tasks(struct doca_dma *dma,
uint32_t *max_num_tasks);
doca_error_t doca_dma_cap_task_memcpy_get_max_buf_size(const struct
doca_devinfo *devinfo, uint64_t *buf_size);
```

## Core Capability

Like any other DOCA library, each DOCA Core module also exposes capability APIs.

For example:

- A hotplug (of emulated PCIe functions) oriented application can check if a specific DOCA Device information structure enables hotplugging emulated devices, by calling:

```
doca_error_t
doca_devinfo_cap_is_hotplug_manager_supported(const struct
doca_devinfo *devinfo, uint8_t *is_hotplug_manager);
```

- An application that works with DOCA mmap to be shared between the host and BlueField, must export the `doca_mmap` from the host and import it from BlueField. Before starting the workflow, the application can check if those operations are supported for a given a `doca_devinfo` using the following APIs:

```
doca_error_t doca_mmap_cap_is_export_pci_supported(const struct
doca_devinfo *devinfo, uint8_t *mmap_export);
doca_error_t
doca_mmap_cap_is_create_from_export_pci_supported(const struct
doca_devinfo *devinfo, uint8_t *from_export);
```

---

# Debuggability

## Return value

All DOCA APIs return the status in the form of [doca\\_error\\_t](#).

The return value of every call to the DOCA API should be checked to verify that it was successful. In case of an error, one should look at the meaning of the returned value in the description of the failing function.

## SDK log

DOCA SDK supports error message and debug prints.

For enabling the DOCA SDK log messages one should create a backend and set the verbosity level of that backend, if needed.

For more details about DOCA log, see section "[DOCA Log](#)".

### 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. PDF Generated on 06/04/2025