DOCA Log

Table of contents

DOCA logging infrastructure allows printing DOCA SDK library error messages, and printing debug and error messages from applications.

To work with the DOCA logging mechanism, the header file `doca_log.h` must be included in every source code using it.

Log Verbosity Level Enumerations

The following verbosity levels are supported by the DOCA logging:

```
enum doca_log_level {
        DOCA_LOG_LEVEL_DISABLE = 10,      /**< Disable log messages */
        DOCA_LOG_LEVEL_CRIT = 20,              /**< Critical log level */
        DOCA_LOG_LEVEL_ERROR = 30,             /**< Error log level */
        DOCA_LOG_LEVEL_WARNING = 40,      /**< Warning log level */
        DOCA_LOG_LEVEL_INFO = 50,              /**< Info log level */
        DOCA_LOG_LEVEL_DEBUG = 60,             /**< Debug log level */
        DOCA_LOG_LEVEL_TRACE = 70,             /**< Trace log level */
};
```

> (i) **Note**
>
> The `DOCA_LOG_LEVEL_TRACE` verbosity level is available only if the macro `DOCA_LOGGING_ALLOW_TRACE` is set before the compilation.

See `doca_log.h` for more information.

Logging Backends

DOCA's logging backend is the target to which log messages are directed.

The following backend types are supported:

- FILE * – file stream which can be any open file or stdout/stderr

- file descriptor – any file descriptor that the system supports, including (but not limited to) raw files, sockets, and pipes

- buf – memory buffer (address and size) that can hold a single message and a callback to be called for every logged message

- syslog – system standard logging

Every logger is created with the following default lower and upper verbosity levels:

- Lower level – `DOCA_LOG_LEVEL_INFO`

- Upper level – `DOCA_LOG_LEVEL_CRIT`

SDK and application logging have different default configuration values and can be controlled separately using the appropriate API.

Every message is printed to every created backend if its verbosity level allows it.

Enabling DOCA SDK Libraries Logging

The DOCA SDK libraries print debug and error messages to all the backends created using the following functions:

- `doca_log_backend_create_with_file_sdk()`

- `doca_log_backend_create_with_fd_sdk()`

- `doca_log_backend_create_with_buf_sdk()`

- `doca_log_backend_create_with_syslog_sdk()`

A newly created SDK backend verbosity level is set to the SDK global verbosity level value. This value can be changed using `doca_log_level_set_global_sdk_limit()`.

`doca_log_level_set_global_sdk_limit()` sets the verbosity level for all existing SDK backends and sets the SDK global verbosity level.

`doca_log_backend_set_sdk_level()` sets the verbosity level of a specific SDK backend.

`doca_log_level_get_global_sdk_limit()` gets the SDK global verbosity level.

> **ⓘ Note**
>
> Messages may change between different versions of DOCA. Users cannot rely on message permanence or formatting.

Enabling DOCA Application Logging

Any source code that uses DOCA can use DOCA logging infrastructure.

Every debug and error messages is printed to all backends created using the following functions:

- `doca_log_backend_create_with_file()`

- `doca_log_backend_create_with_fd()`

- `doca_log_backend_create_with_buf()`

- `doca_log_backend_create_with_syslog()`

The lower and upper levels of a newly created backend are set to the default values. Those values can be changed using `doca_log_backend_set_level_lower_limit()` and `doca_log_backend_set_level_upper_limit()`.

`doca_log_backend_create_standard()` creates a default non-configurable set of two backends:

- `stdout` prints the range from a global minimum level up to `DOCA_LOG_LEVEL_INFO`

- `stderr` prints the range from `DOCA_LOG_LEVEL_WARNING` level up to `DOCA_LOG_LEVEL_CRIT`

`doca_log_backend_set_level_lower_limit_strict()` marks the lower log level limit of a backend as strict, preventing it from being lowered by future log level changes. It

is both global and direct.

`doca_log_backend_set_level_upper_limit_strict()` marks the upper log level limit of a backend as strict, preventing it from being raised by future log level changes. It is both global and direct.

`doca_log_level_set_global_lower_limit()` sets the lower limit for all existing backends not marked as strict and sets the global application lower limit.

`doca_log_level_set_global_upper_limit()` sets the upper limit for all existing backends not marked as strict and sets the global application upper limit.

Logging DOCA Application Messages

To use the DOCA logging infrastructure with your source code to log its messages, users must call, at the beginning of the file, the macro `DOCA_LOG_REGISTER(source)` just before using the DOCA logging functionality. This macro handles the registration and the teardown from the DOCA logging.

Printing a message can be done by calling one of the following macros (with the same usage as `printf()`):

- `DOCA_LOG_CRIT(format, ...)`

- `DOCA_LOG_ERR(format, ...)`

- `DOCA_LOG_WARN(format, ...)`

- `DOCA_LOG_INFO(format, ...)`

- `DOCA_LOG_DBG(format, ...)`

- `DOCA_LOG_TRC(format, ...)`

The message is printed to all the application's backends with configured lower and upper logging limits.

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.