



# **DOCA File Compression Application Guide**

# Table of contents

Introduction

---

System Design

---

Application Architecture

---

DOCA Libraries

---

Compiling the Application

---

Compiling All Applications

---

Compiling File Compression Application Only

---

Troubleshooting

---

Running the Application

---

Application Execution

---

Command Line Flags

---

Troubleshooting

---

Application Code Flow

---

References

---

This document provides a file compression implementation on top of the NVIDIA® BlueField® DPU.

## Introduction

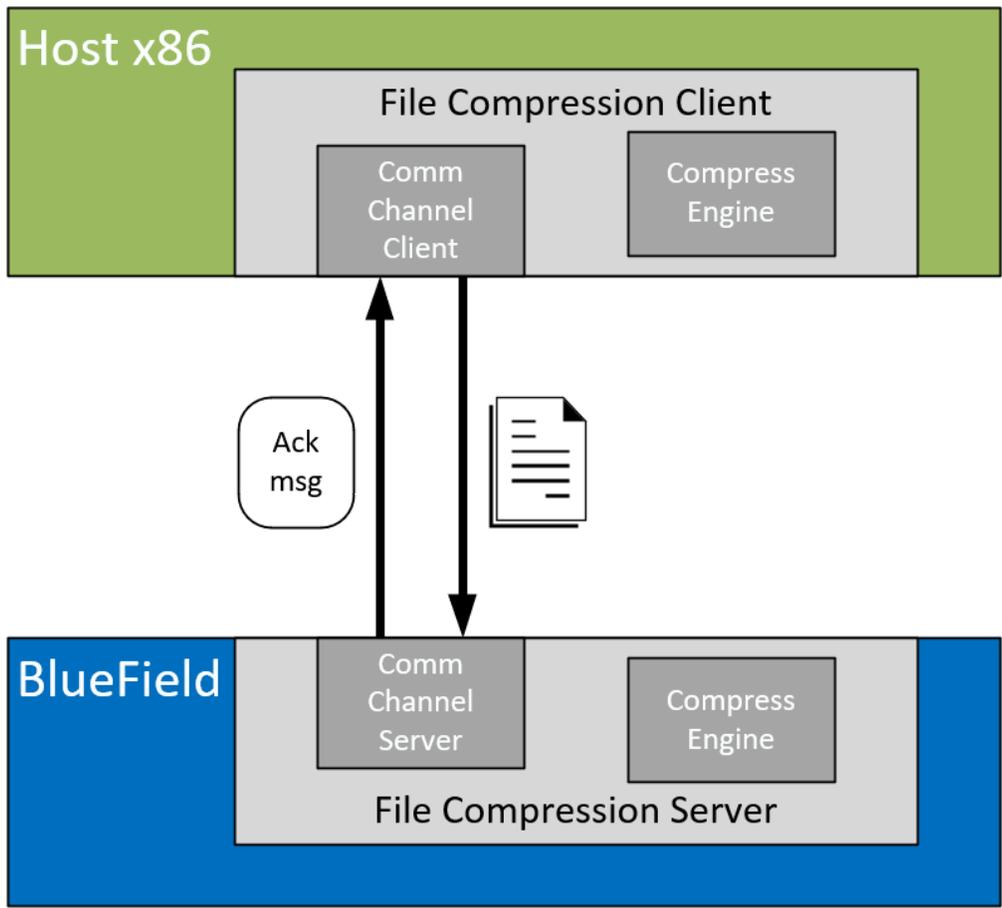
The file compression application exhibits how to use the [DOCA Compress](#) API to compress and decompress data using hardware acceleration as well as sending and receiving it using the [DOCA Comch](#) API.

The application's logic includes both a client and a server:

- Client side – the application opens a file, compresses it, and sends the checksum of the source file with the compressed data to the server
- Server side – the application saves the received file in a buffer, decompresses it, and compares the received checksum with the calculated one

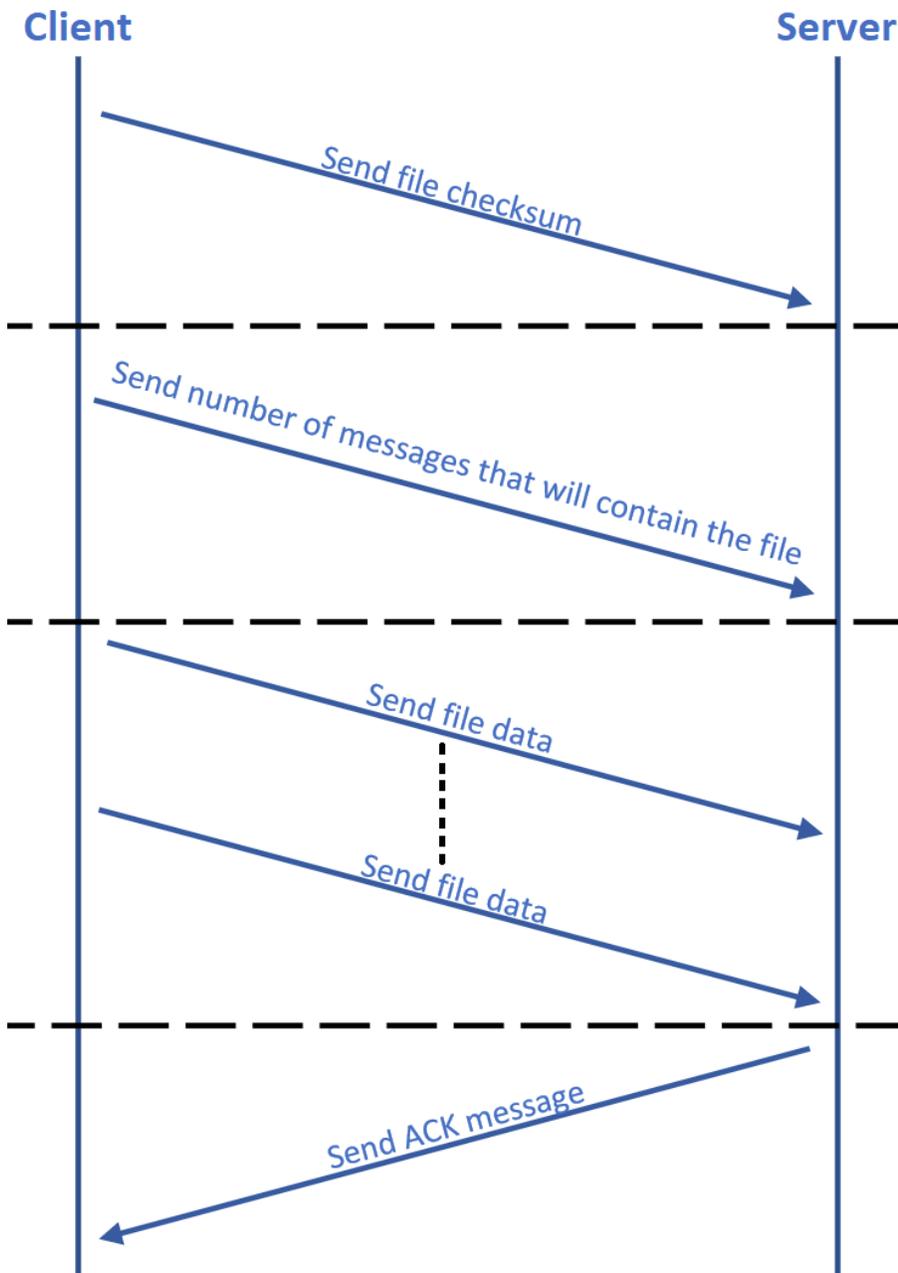
## System Design

The file compression application client runs on the host and the server runs on the DPU.



## Application Architecture

The file compression application runs on top of the DOCA Comm Channel API to send and receive the file from the host and to the DPU.



1. Connection is established on both sides by DOCA Comm Channel API.
2. Client compresses the data:
  - When compress engine is available – submits compress job with DOCA Compress API and sends the result to the server
  - When compress engine is unavailable – compresses the data in software
3. Client sends the number of messages needed to send the compressed content of the file.

4. Client sends data segments in size of up to 4080 bytes.
5. Server saves the received data in a buffer and submits a decompress job.
6. Server sends an ACK message to the client when all parts of the file are received successfully.
7. Server compares the received checksum to the calculated checksum.
8. Server writes the decompressed data to an output file.

## DOCA Libraries

This application leverages the following DOCA libraries:

- [DOCA Compress](#)
- [DOCA Comch](#)

Refer to their respective programming guide for more information.

## Compiling the Application

### Info

Please refer to the [DOCA Installation Guide for Linux](#) for details on how to install BlueField-related software.

The installation of DOCA's reference applications contains the sources of the applications, alongside the matching compilation instructions. This allows for compiling the applications "as-is" and provides the ability to modify the sources, then compile a new version of the application.

### Tip

For more information about the applications as well as development and compilation tips, refer to the [DOCA Reference Applications](#) page.

The sources of the application can be found under the application's directory:

```
/opt/mellanox/doca/applications/file_compression/.
```

## Compiling All Applications

All DOCA applications are defined under a single meson project. So, by default, the compilation includes all of them.

To build all the applications together, run:

```
cd /opt/mellanox/doca/applications/  
meson /tmp/build  
ninja -C /tmp/build
```

### Info

```
doca_file_compression is created under  
/tmp/build/file_compression/.
```

## Compiling File Compression Application Only

To directly build only the file compression application:

```
cd /opt/mellanox/doca/applications/
```

```
meson /tmp/build -Denable_all_applications=false -  
Denable_file_compression=true  
ninja -C /tmp/build
```

### **i** Info

`doca_file_compression` is created under  
`/tmp/build/file_compression/`.

Alternatively, the user may set the desired flags in the `meson_options.txt` file instead of providing them in the compilation command line:

1. Edit the following flags in

`/opt/mellanox/doca/applications/meson_options.txt`:

- Set `enable_all_applications` to `false`
- Set `enable_file_compression` to `true`

2. Run the following compilation commands :

```
cd /opt/mellanox/doca/applications/  
meson /tmp/build  
ninja -C /tmp/build
```

### **i** Info

`doca_file_compression` is created under  
`/tmp/build/file_compression/`.

# Troubleshooting

Refer to the [DOCA Troubleshooting](#) for any issue encountered with the compilation of the application.

## Running the Application

### Application Execution

The file compression application is provided in source form. Therefore, a compilation is required before the application can be executed.

1. Application usage instructions:

```
Usage: doca_file_compression [DOCA Flags] [Program Flags]

DOCA Flags:
  -h, --help                Print a help synopsis
  -v, --version             Print program version
information
  -l, --log-level           Set the (numeric) log
level for the program <10=DISABLE, 20=CRITICAL, 30=ERROR,
40=WARNING, 50=INFO, 60=DEBUG, 70=TRACE>
  --sdk-log-level          Set the SDK (numeric) log
level for the program <10=DISABLE, 20=CRITICAL, 30=ERROR,
40=WARNING, 50=INFO, 60=DEBUG, 70=TRACE>
  -j, --json <path>       Parse all command flags
from an input json file

Program Flags:
  -p, --pci-addr           DOCA Comm Channel device
PCI address
```

```
-r, --rep-pci          DOCA Comm Channel device
representor PCI address
-f, --file            File to send by the
client / File to write by the server
-t, --timeout        Application timeout for
receiving file content messages, default is 5 sec
```

### **(i) Info**

This usage printout can be printed to the command line using the `-h` (or `--help`) options:

```
./doca_file_compression -h
```

### **(i) Info**

For additional information, refer to section "[Command Line Flags](#)".

2. CLI example for running the application on BlueField:

```
./doca_file_compression -p 03:00.0 -r 3b:00.0 -f received.txt
```

### **(i) Note**

Both the DOCA Comm Channel device PCIe address (`03:00.0`) and the DOCA Comm Channel device representor PCIe address (

3b:00.0) should match the addresses of the desired PCIe devices.

3. CLI example for running the application on the host:

```
./doca_file_compression -p 3b:00.0 -f send.txt
```

**Note**

The DOCA Comm Channel device PCIe address (3b:00.0) should match the address of the desired PCIe device.

4. The application also supports a JSON-based deployment mode, in which all command-line arguments are provided through a JSON file:

```
./doca_file_compression --json [json_file]
```

For example:

```
./doca_file_compression --json ./file_compression_params.json
```

**Note**

Before execution, ensure that the used JSON file contains the correct configuration parameters, and especially the PCIe addresses necessary for the deployment.

## Command Line Flags

Flag Type	Short Flag	Long Flag/JSON Key	Description	JSON Content
General flags	<code>h</code>	<code>help</code>	Prints a help synopsis	N/A
	<code>v</code>	<code>version</code>	Prints program version information	N/A
	<code>l</code>	<code>log-level</code>	Set the log level for the application: <ul style="list-style-type: none"> <li>• DISABLE=10</li> <li>• CRITICAL=20</li> <li>• ERROR=30</li> <li>• WARNING=40</li> <li>• INFO=50</li> <li>• DEBUG=60</li> <li>• TRACE=70 (requires compilation with <code>TRACE</code> log level support)</li> </ul>	<pre>"log-level" : 60</pre>
	N/A	<code>sdk-log-level</code>	Sets the log level for the program: <ul style="list-style-type: none"> <li>• DISABLE=10</li> <li>• CRITICAL=20</li> <li>• ERROR=30</li> <li>• WARNING=40</li> <li>• INFO=50</li> <li>• DEBUG=60</li> <li>• TRACE=70</li> </ul>	<pre>"sdk-log- level" : 40</pre>
	<code>j</code>	<code>json</code>	Parse all command flags from an input JSON file	N/A

Flag Type	Short Flag	Long Flag/JSON Key	Description	JSON Content
Program flags	f	file	<p>For client – path to the file to be sent For server – path to write the file into</p> <p><b>Note</b> This is a mandatory flag.</p>	<pre>"file" : "/tmp/data. txt"</pre>
	p	pci-addr	<p>Comm Channel DOCA device PCIe address</p> <p><b>Note</b> This is a mandatory flag.</p>	<pre>"pci-addr" : 03 :00.1</pre>
	r	rep-pci	<p>Comm Channel DOCA device representor PCIe address</p> <p><b>Note</b> This flag is mandatory only on the DPU.</p>	<pre>"rep-pci" : b1 :00.1</pre>

**i Info**

Refer to [DOCA Arg Parser](#) for more information regarding the supported flags and execution modes.

# Troubleshooting

Refer to the [DOCA Troubleshooting](#) for any issue encountered with the installation or execution of the DOCA applications.

## Application Code Flow

1. Parse application argument.

1. Initialize arg parser resources and register DOCA general parameters.

```
doca_argp_init();
```

2. Register file compression application parameters.

```
register_file_compression_params();
```

3. Parse the arguments.

```
doca_argp_start();
```

1. Parse app parameters.

2. Set endpoint attributes.

```
set_endpoint_properties();
```

1. Set maximum message size of 4080 bytes.

2. Set maximum number of messages allowed.

3. Create comm channel endpoint.

```
doca_comm_channel_ep_create();
```

1. Create endpoint for client/server.

4. Run client/server main logic.

```
file_compression_client/server();
```

5. Clean up the file compression application.

```
file_compression_cleanup();
```

1. Free all application resources.

6. Arg parser destroy.

```
doca_argp_destroy();
```

## References

- `/opt/mellanox/doca/applications/file_compression/`
- `/opt/mellanox/doca/applications/file_compression/file_compression`

**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 05/05/2025