



DPDK

DATA PLANE DEVELOPMENT KIT

Compression Device Drivers

Release 18.05.1

September 07, 2018

CONTENTS

1	Compression Device Supported Functionality Matrices	1
1.1	Supported Feature Flags	1
2	ISA-L Compression Poll Mode Driver	2
2.1	Features	2
2.2	Limitations	3
2.3	Installation	3
2.4	Initialization	4

COMPRESSION DEVICE SUPPORTED FUNCTIONALITY MATRICES

1.1 Supported Feature Flags

Table 1.1: Features availability in compression drivers

Feature	i s a l
HW Accelerated	
CPU SSE	Y
CPU AVX	Y
CPU AVX2	Y
CPU AVX512	Y
CPU NEON	
Stateful	
By-Pass	
Chained mbufs	
Deflate	Y
LZS	
Adler32	
Crc32	
Adler32&Crc32	
Fixed	Y
Dynamic	Y

ISA-L COMPRESSION POLL MODE DRIVER

The ISA-L PMD (**librte_pmd_isal_comp**) provides poll mode compression & decompression driver support for utilizing Intel ISA-L library, which implements the deflate algorithm for both Deflate(compression) and Inflate(decompression).

2.1 Features

ISA-L PMD has support for:

Compression/Decompression algorithm:

- DEFLATE

Huffman code type:

- FIXED
- DYNAMIC

Window size support:

- 32K

Level guide:

The ISA-L levels have been mapped to somewhat correspond to the same ZLIB level, i.e. ZLIB L1 gives a compression ratio similar to ISA-L L1. Compressdev level 0 enables “No Compression”, which passes the uncompressed data to the output buffer, plus deflate headers. The ISA-L library does not support this, therefore compressdev level 0 is not supported.

The compressdev API has 10 levels, 0-9. ISA-L has 4 levels of compression, 0-3. As a result the level mappings from the API to the PMD are shown below.

Table 2.1: Level mapping from Compressdev to ISA-L PMD.

Compressdev API Level	PMD Functionality	Internal ISA-L Level
0	No compression, Not Supported	—
1	Dynamic (Fast compression)	1
2	Dynamic (Higher compression ratio)	2
3	Dynamic (Best compression ratio)	3 (Level 2 if no AVX512/AVX2)
4	Dynamic (Best compression ratio)	Same as above
5	Dynamic (Best compression ratio)	Same as above
6	Dynamic (Best compression ratio)	Same as above
7	Dynamic (Best compression ratio)	Same as above
8	Dynamic (Best compression ratio)	Same as above
9	Dynamic (Best compression ratio)	Same as above

Note: The above table only shows mapping when API calls for dynamic compression. For fixed compression, regardless of API level, internally ISA-L level 0 is always used.

2.2 Limitations

- Chained mbufs will not be supported until a future release, meaning max data size being passed to PMD through a single mbuf is 64K - 1. If data is larger than this it will need to be split up and sent as multiple operations.
- Compressdev level 0, no compression, is not supported.
- Checksums will not be supported until future release.

2.3 Installation

- To build DPDK with Intel's ISA-L library, the user is required to download the library from <https://github.com/01org/isa-l>.
- Once downloaded, the user needs to build the library, the ISA-L autotools are usually sufficient:

```
./autogen.sh
./configure
```

- make can be used to install the library on their system, before building DPDK:

```
make
sudo make install
```

- To build with meson, the **libisal.pc** file, must be copied into "pkgconfig", e.g. /usr/lib/pkgconfig or /usr/lib64/pkgconfig depending on your system, for meson to find the ISA-L library. The **libisal.pc** is located in library sources:

```
cp isal/libisal.pc /usr/lib/pkgconfig/
```

2.4 Initialization

In order to enable this virtual compression PMD, user must:

- Set `CONFIG_RTE_LIBRTE_PMD_ISAL=y` in `config/common_base`.

To use the PMD in an application, user must:

- Call `rte_vdev_init("compress_isal")` within the application.
- Use `--vdev="compress_isal"` in the EAL options, which will call `rte_vdev_init()` internally.

The following parameter (optional) can be provided in the previous two calls:

- `socket_id`: Specify the socket where the memory for the device is going to be allocated (by default, `socket_id` will be the socket where the core that is creating the PMD is running on).