

# Filters

## LZO Filter

**Filter ID:** 305

**Filter Description:**

- LZO is a portable lossless data compression library written in ANSI C.
- Reliable and thoroughly tested. High adoption - each second terrabytes of data are compressed by LZO. No bugs since the first release back in 1996.
- Offers pretty fast compression and \*extremely\* fast decompression.
- Includes slower compression levels achieving a quite competitive compression ratio while still decompressing at this very high speed.
- Distributed under the terms of the GNU General Public License (GPL v2+). Commercial licenses are available on request.
- Military-grade stability and robustness.

**Links:**

<http://www.oberhumer.com/opensource/lzo/>

<http://www.pytables.org>

**Contact Information:**

Francesc Alted

Email: faltet at pytables dot org

---

## BZIP2 Filter

**Filter ID:** 307

**Filter Description:**

bzip2 is a freely available, patent free, high-quality data compressor. It typically compresses files to within 10% to 15% of the best available techniques (the PPM family of statistical compressors), whilst being around twice as fast at compression and six times faster at decompression.

**Links:**

<http://www.bzip.org>

<http://www.pytables.org>

**Contact Information:**

Francesc Alted

Email: faltet at pytables dot org

---

## LZF Filter

**Filter ID:** 32000

**Filter Description:**

The LZF filter is an alternative DEFLATE-style compressor for HDF5 datasets, using the free LZF library by Marc Alexander Lehmann. Its main benefit over the built-in HDF5 DEFLATE filter is speed; in memory-to-memory operation as part of the filter pipeline, it typically compresses 3x-5x faster than DEFLATE, and decompresses 2x faster, while maintaining 50% to 90% of the DEFLATE compression ratio.

LZF can be used to compress any data type, and requires no compile-time or run-time configuration. HDF5 versions 1.6.5 through 1.8.3 are supported. The filter is written in C and can be included directly in C or C++ applications; it has no external dependencies. The license is 3-clause BSD (virtually unrestricted, including commercial applications).

More information, downloads, and benchmarks, are available at the <http://h5py.org/lzf/>.

**Additional Information:**

The LZF filter was developed as part of the h5py project, which implements a general-purpose interface to HDF5 from Python.

**Links:**

The h5py homepage: <http://h5py.org>

The LZF library homepage: <http://home.schmorp.de/marc/liblzf.html>

**Contact Information:**

Andrew Collette  
Web: <http://h5py.org>

---

**Blosc Filter**

**Filter ID:** 32001

**Filter Description:**

Blosc is a high performance compressor optimized for binary data. It has been designed to compress data very fast, at the expense of achieving lesser compression ratios than, say, zlib+shuffle. It is mainly meant to not introduce a significant delay when dealing with data that is stored in high-performance I/O systems (like large RAID cabinets, or even the OS filesystem memory cache).

It uses advanced cache-efficient techniques to reduce activity on the memory bus as much as possible. It also leverages SIMD (SSE2) and multi-threading capabilities present in nowadays multi-core processors so as to accelerate the compression/decompression process to a maximum.

**Links:**

<http://blosc.org/>  
<http://www.pytables.org>

**Contact Information:**

Francesc Alted  
Email: [faltet@pytables.org](mailto:faltet@pytables.org)

---

**MAFISC Filter**

**Filter ID:** 32002

**Filter Description:**

This compressing filter exploits the multidimensionality and smoothness characterizing many scientific data sets. It adaptively applies some filters to preprocess the data and uses lzma as the actual compression step. It significantly outperforms pure lzma compression on most datasets.

The software is currently under a rather unrestrictive two clause BSD style license.

**Links:**

<http://wr.informatik.uni-hamburg.de/research/projects/icomex/mafisc>

**Contact Information:**

Nathanael Huebbe  
Email: [nathanael.huebbe@informatik.uni-hamburg.de](mailto:nathanael.huebbe@informatik.uni-hamburg.de)

---

**Snappy Filter**

**Filter ID:** 32003

**Filter Description:**

Snappy is a compression/decompression library. It does not aim for maximum compression, or compatibility with any other compression library; instead, it aims for very high speeds and reasonable compression. For instance, compared to the fastest mode of zlib, Snappy is an order of magnitude faster for most inputs, but the resulting compressed files are anywhere from 20% to 100% bigger. On a single core of a Core i7 processor in 64-bit mode, Snappy compresses at about 250 MB/sec or more and decompresses at about 500 MB/sec or more.

Snappy is widely used inside Google, in everything from BigTable and MapReduce to our internal RPC systems. (Snappy has previously been referred to as .Zippy. in some presentations and the likes.)

**Links:**

<http://google.github.io/snappy/>

**Contact Information:**

Michael Rissi (Dectris Ltd.)  
Email: [michael.rissi@dectris.com](mailto:michael.rissi@dectris.com)

---

## LZ4 Filter

**Filter ID:** 32004

**Filter Description:**

LZ4 is a very fast lossless compression algorithm, providing compression speed at 300 MB/s per core, scalable with multi-cores CPU. It also features an extremely fast decoder, with speeds up and beyond 1GB/s per core, typically reaching RAM speed limits on multi-core systems. For a format description of the LZ4 compression filter in HDF5, see [HDF5\\_LZ4.pdf](#).

**Links:**

LZ4 Algorithm: <https://github.com/nexusformat/HDF5-External-Filter-Plugins/tree/master/LZ4>

LZ4 Code:

Although the LZ4 software is not supported by The HDF Group, it is included in The HDF Group SVN repository so that it can be tested regularly with HDF5. For convenience, users can obtain it from SVN with the following command:

```
svn checkout https://svn.hdfgroup.org/hdf5_plugins/trunk/LZ4 LZ4
```

**Contact Information:**

Michael Rissi (Dectris Ltd.)  
Email: michael dot rissi at dectris dot com

---

## APAX

**Filter ID:** 32005

Appears to be no longer available

---

## CBF

**Filter ID:** 32006

**Filter Description:**

All imgCIF/CBF compressions and decompressions, including Canonical, Packed, Packed Version 2, Byte Offset and Nibble Offset.  
License Information: GPL and LGPL

**Contact Information:**

Herbert J. Bernstein  
Email: yayahjb at gmail dot com

---

## JPEG-XR

**Filter Id:** 32007

**Filter Description:**

Filter that allows HDF5 image datasets to be compressed or decompressed using the JPEG-XR compression method.

**Links:**

[JPEG-XR Compression Method](#)  
[JPEG-XR Filter for HDF5](#)

**Contact Information:**

Marvin Albert  
Email: marvin dot albert at gmail dot com

---

## bitshuffle

**Filter Id:** 32008

**Filter Description:**

This filter shuffles data at the bit level to improve compression. CHIME uses this filter for data acquisition.

**Links:**

[bitshuffle](#)  
[CHIME](#)

**Contact Information:**

Kiyoshi Masui  
Email: kiyo at physics dot ubc dot ca

---

## SPDP

**Filter Id:** 32009

**Filter Description:**

SPDP is a fast, lossless, unified compression/decompression algorithm designed for both 32-bit single-precision (float) and 64-bit double-precision (double) floating-point data. It also works on other data.

**Link to the filter:**

<http://cs.txstate.edu/~burtscher/research/SPDP/>

**Contact Information:**

Martin Burtscher  
Email: burtscher at txstate dot edu

---

## LPC-Rice

**Filter Id:** 32010

**Filter Description:**

LPC-Rice is a fast lossless compression codec that employs Linear Predictive Coding together with Rice coding. It supports multi-threading and SSE2 vector instructions, enabling it to exceed compression and decompression speeds of 1 GB/s.

**Link to the filter:**

<https://sourceforge.net/projects/lprrice/>

**Contact Information:**

Frans van den Bergh  
Email: fvdbergh at csir dot co dot za

Derick Swanepoel  
Email: dswanepoel at gmail dot com

---

## CCSDS-123

**Filter Id:** 32011

**Filter Description:**

CCSDS-123 is a multi-threaded HDF5 compression filter using the ESA CCSDS-123 implementation.

**Link to the filter:**

<https://sourceforge.net/projects/ccsds123-hdf-filter/>

**Contact Information:**

Frans van den Bergh  
Email: fvdbergh at csir dot co dot za

Derick Swanepoel  
Email: dswanepoel at gmail dot com

---

---

## JPEG-LS

**Filter Id:** 32012

**Filter Description:**

JPEG-LS is a multi-threaded HDF5 compression filter using the CharLS JPEG-LS implementation.

**Link to the filter:**

<https://sourceforge.net/projects/jpegls-hdf-filter/>

**Contact Information:**

Frans van den Bergh  
Email: fvdbergh at csir dot co dot za

Derick Swanepoel  
Email: dswanepoel at gmail dot com

---

## zfp

**Filter Id:** 32013

**Filter Description:**

zfp is a BSD licensed open source C++ library for compressed floating-point arrays that support very high throughput read and write random access. zfp was designed to achieve high compression ratios and therefore uses **lossy** but optionally error-bounded compression. Although bit-for-bit lossless compression is not always possible, zfp is usually accurate to within machine epsilon in near-lossless mode, and is often orders of magnitude more accurate and faster than other lossy compressors.

**Link to the filter:**

<https://github.com/LLNL/H5Z-ZFP>

For more information see: <http://computation.llnl.gov/projects/floating-point-compression/>

**Contact Information:**

Mark Miller  
Email: miller86 at llnl dot gov

Peter Lindstrom  
Email: pl at llnl dot gov

---

## fpzip

**Filter Id:** 32014

**Filter Description:**

fpzip is a library for lossless or lossy compression of 2D or 3D floating-point scalar fields. Although written in C++, fpzip has a C interface. fpzip was developed by Peter Lindstrom at LLNL.

**Link to the filter:**

For more information see: <http://computation.llnl.gov/projects/floating-point-compression/>

**Contact Information:**

Peter Lindstrom  
Email: pl at llnl dot gov

---

## Zstandard

**Filter Id:** 32015

**Filter Description:**

Zstandard is a real-time compression algorithm, providing high compression ratios. It offers a very wide range of compression / speed trade-offs, while being backed by a very fast decoder. The [Zstandard library](#) is provided as open source software using a BSD license.

**Link to the filter:**

<https://github.com/aparamon/HDF5Plugin-Zstandard>

**Contact Information:**

Andrey Paramonov  
Email: paramon at acdlabs dot ru

---

## B<sup>3</sup>D

**Filter Id:** 32016

**Filter Description:**

B<sup>3</sup>D is a fast (~1 GB/s), GPU based image compression method, developed for light-microscopy applications. Alongside lossless compression, it offers a noise dependent lossy compression mode, where the loss can be tuned as a proportion of the inherent image noise (accounting for photon shot noise and camera read noise). It not only allows for fast compression during image, but can achieve compression ratios up 100.

**Information:**

<http://www.biorxiv.org/content/early/2017/07/21/164624>

---

## SZ

**Filter Id:** 32017

**Filter Description:**

SZ is a fast and efficient error-bounded lossy compressor for floating-point data. It was developed for scientific applications producing large-scale HPC data sets. SZ supports C, Fortran, and Java and has been tested on Linux and Mac OS X.

**Link to the filter:**

[Information](#)  
[github](#)  
[License](#)

**Contact Information:**

Sheng Di  
Email: sdi1 at anl dot gov

Franck Cappello  
Email: cappello at mcs dot anl dot gov

---

## FCIDECOMP

**Filter Id:** 32018

**Filter Description:**

FCIDECOMP is a third-party compression filter used at EUMETSAT for the compression of netCDF-4 files. It is a codec implementing JPEG-LS using CharLS used for satellite imagery.

**Link to the filter:**

All software and documentation can be found at this link: [ftp://ftp.eumetsat.int/pub/OPS/out/test-data/Test-data-for-External-Users/MTG\\_FCI\\_L1c\\_Compressed-Datasets\\_and\\_Decompression-Plugin\\_April2017/Decompression\\_Plugin/](ftp://ftp.eumetsat.int/pub/OPS/out/test-data/Test-data-for-External-Users/MTG_FCI_L1c_Compressed-Datasets_and_Decompression-Plugin_April2017/Decompression_Plugin/)

**Contact Information:**

Dr. Daniel Lee  
Email: daniel dot lee at eumetsat dot int

---

## JPEG

**Filter Id:** 32019

**Filter Description:**

This is a lossy compression filter. It provides a user-specified "quality factor" to control the trade-off of size versus accuracy.

**Link to the filter:**

[Information](#)  
[Github](#)  
[License](#)

libjpeg: This library is available as a package for most Linux distributions, and source code is available from <https://www.ijg.org/>.

**Restrictions:**

- Only 8-bit unsigned data arrays are supported.
- Arrays must be either:
  - 2-D monochromatic [NumColumns, NumRows]
  - 3-D RGB [3, NumColumns, NumRows]
- Chunking must be set to the size of one entire image so the filter is called once for each image.

**Using the JPEG filter in your application:**

HDF5 only supports compression for "chunked" datasets; this just means that you need to call H5Pset\_chunk to specify a chunk size. The chunking must be set to the size of a single image for the JPEG filter to work properly.

When calling H5Pset\_filter for compression it must be called with cd\_nelmts=4 and cd\_values as follows:

```
cd_values[0] = quality factor (1-100)
cd_values[1] = numColumns
cd_values[2] = numRows
cd_values[3] = 0=Mono, 1=RGB
```

Common h5repack parameter: UD=32019,0,4,q,c,r,t

**Contact Information:**

Mark Rivers , University of Chicago (rivers at cars.uchicago.edu)

---

## VBZ

**Filter Id:** 32020

**Filter Description:**

This filter is used by Oxford Nanopore specifically to compress raw dna signal data (signed integer). To achieve this it uses both:

- streamvbyte (<https://github.com/lemire/streamvbyte>)
- zstd (<https://github.com/facebook/zstd>)

**Link to the filter:**

**Contact Information:**

George Pimm

---

## FAPEC

**Filter Id:** 32021

**Filter Description:**

FAPEC is a versatile and efficient data compressor, initially designed for satellite payloads but later extended for ground applications. It relies on an outlier-resilient entropy coding core with similar ratios and speeds than CCSDS 121.0 (adaptive Rice).

FAPEC has a large variety of pre-processing stages and options: images (greyscale, colour, hyperspectral); time series or waveforms (including interleaving, e.g. for multidimensional or interleaved time series or tabular data); floating point (single+double precision); text (including LZW compression and our faster FAPECLZ); tabulated text (CSV); genomics (FastQ); geophysics (Kongsberg's water column datagrams); etc.

Most stages support samples of 8 to 24 bits (big/little endian, signed/unsigned), and lossless/lossy options. It can be extended with new, tailored

pre-processing stages. It includes encryption options (AES-256 based on OpenSSL, and our own XXTEA implementation).

The FAPEC library and CLI runs on Linux, Windows and Mac. The HDF5 user must request and install the library separately, thus allowing to upgrade it without requiring changes in your HDF5 code.

**Link to relevant information including licensing information:**

<https://www.dapcom.es/fapec/>

<https://www.dapcom.es/get-fapec/>

[https://www.dapcom.es/resources/FAPEC\\_EndUserLicenseAgreement.pdf](https://www.dapcom.es/resources/FAPEC_EndUserLicenseAgreement.pdf)

**Contact Information:**

Jordi Portell i de Mora (DAPCOM Data Services S.L.)

fapec at dapcom dot es