

# Registered Filter Plugins

Please be sure to see [HDF5 Filter Plugins](#), a convenience software that packages together many of the commonly used filters that users have created and registered.

## Information on Registered Filter Plugins

Members of the HDF5 user community can create and register Third-Party (compression or other) filters for use with HDF5. See [Example Code to Enable BZIP2 Compression in HDF5](#) for how to create a filter.

To register a filter please contact [The HDF Helpdesk](#) with the following information:

- Contact information for the developer requesting a new identifier
- Short description of the new filter
- Links to any relevant information including licensing information

Here is the current policy regarding filter identifier assignment:

The filter identifier is designed to be a unique identifier for the filter. Values from zero through 32,767 are reserved for filters supported by The HDF Group in the HDF5 library and for filters requested and supported by the 3rd party.

Values from 32768 to 65535 are reserved for non-distributed uses (e.g., internal company usage) or for application usage when testing a feature. The HDF Group does not track or document the usage of filters with identifiers from this range.

Please contact the maintainer of a filter for help with the filter/compression support in HDF5.

## List of Filters Registered with The HDF Group

Filter Identifier	Name	Short Description
305	LZO	LZO lossless compression used by PyTables
307	BZIP2	BZIP2 lossless compression used by PyTables
32000	LZF	LZF lossless compression used by H5Py project
32001	BLOSC	Blosc lossless compression used by PyTables
32002	MAFISC	Modified LZMA compression filter, MAFISC (Multidimensional Adaptive Filtering Improved Scientific data Compression)
32003	Snappy	Snappy lossless compression
32004	LZ4	LZ4 fast lossless compression algorithm
32005	APAX	Samplify's APAX Numerical Encoding Technology
32006	CBF	All imgCIF/CBF compressions and decompressions, including Canonical, Packed, Packed Version 2, Byte Offset and Nibble Offset
32007	JPEG-XR	Enables images to be compressed/decompressed with JPEG-XR compression
32008	bitshuffle	Extreme version of shuffle filter that shuffles data at bit level instead of byte level
32009	SPDP	SPDP fast lossless compression algorithm for single- and double-precision floating-point data
32010	LPC-Rice	LPC-Rice multi-threaded lossless compression
32011	CCSDS-123	ESA CCSDS-123 multi-threaded compression filter
32012	JPEG-LS	CharLS JPEG-LS multi-threaded compression filter
32013	zfp	Rate, accuracy or precision bounded compression for floating-point arrays

32014	fpzip	Fast and Efficient Lossy or Lossless Compressor for Floating-Point Data
32015	Zstandard	Real-time compression algorithm with wide range of compression / speed trade-off and fast decoder
32016	B <sup>3</sup> D	GPU based image compression method developed for light-microscopy applications
32017	SZ	An error-bounded lossy compressor for scientific floating-point data
32018	FCIDECOMP	EUMETSAT CharLS compression filter for use with netCDF
32019	JPEG	Jpeg compression filter
32020	VBZ	Compression filter for raw dna signal data used by Oxford Nanopore
32021	FAPEC	Versatile and efficient data compressor supporting many kinds of data and using an outlier-resilient entropy coder

## Example Code to enable BZIP2 Compression in HDF5

The [bz\\_example.tar.gz](#) file contains an example of implementing the BZIP2 filter to enable BZIP2 compression in HDF5. (This example is based on PyTables code that uses BZIP2 compression.). Download and uncompress this file as follows:

```
gzip -cd bz_example.tar.gz | tar xvf -
```

To compile the example, you will need to install the HDF5 library and use the h5cc compile script found in the bin/ directory of the HDF5 installation. For information on obtaining HDF5, see: [Obtain the Latest HDF5 Software](#)

For information on h5cc, see: [Compiling Your HDF5 Application](#)

Please note that tools like h5dump that display information in an HDF5 file will not be able to display data that is compressed with BZIP2 compression, since BZIP2 is not implemented in HDF5.

However, as of HDF5-1.8.11, a new HDF5 feature will enable the h5dump tool to determine that the data is compressed with an external compression filter such as BZIP2, and will automatically load the appropriate library and display the uncompressed data.

The bz\_example example code can be used for modifying the HDF5 source to "include" BZIP2 as one of the "internal" filters. For information on how to do this, see how ZLIB (the deflate filter) is implemented in the HDF5 source code. Specifically look at these files:

H5Z.c, H5Zdeflate.c and H5Pocpl.c