RFC: Registering Data Filters to be used with the HDF5 Library

Quincey Koziol
Elena Pourmal

This document proposes a new policy and a process of registering data filters to be used with the HDF5 library.

1 Introduction

The ability to use different types of compression and other data filters with the HDF5 library is one of the very popular features in HDF5. Compression methods such as Gzip and Szip, N-bit and Scale+Offset were implemented in HDF5 and supported by the HDF5 developers along with a checksum and shuffling filters, while bzip2, LZO and LZF compression methods were implemented and supported by the developers of PyTables and H5Py packages. In general, data stored in HDF5 may effectively become “encrypted” if the filter applied to the data when it was written is not available to the HDF5 library that reads the data.

To mitigate the issue, the HDF5 Development Team provides a policy and registration mechanism for the filters. Unfortunately, current policy doesn’t clearly define the process of getting a filter identifier. Also there is no a central place where an HDF5 user can find information about filters used by the HDF5 team and other developers. Two examples below illustrate the issues:

1. PyTables uses identifiers with value 305 for LZO and value 307 for bzip2 that were intended to be utilized by the HDF5 team.
2. A user may implement bzip2 filter and use a different identifier than PyTables making data unavailable to PyTables.

This document proposes new policy on assigning filter identifiers, registering them with the HDF5 Developers Team and making information about the new filters available to HDF5 users’ community.

2 Current registering mechanism and “Filter identifier assignment” policy

HDF5 provides a registration mechanism to make a filter available to the HDF5 library by assigning an integer number known as a filter identifier along with other information about the filter. For details on a new filter registration with the library we refer a reader to the H5Z interface routines in the HDF5 Reference Manual [1].

The description of a filter identifier and the policy of assigning of a filter identifier is provided in the HDF5 File Format Specification in the “Data Storage – Filter Pipeline” section [2]. It reads:

“This [filter identifier] is a unique (except in the case of testing) identifier for the filter. Values from zero through 255 are reserved for filters defined by the HDF Group's HDF5 library. Values 256 through

511 have been set aside for use when developing/testing new filters. The remaining values are allocated to specific filters by contacting the HDF5 Development Team at help@hdfgroup.org".

HDF5 File Format specification also lists identifiers of all filters supported by the HDF5 Development Team and available in the HDF5 library.

The current policy reserves too small of an interval for the filters that might be used by the HDF5 Development Team. It also doesn't state what kind of information should be provided when an identifier for a new filter is requested, nor what the process is of working with developers who request registration of a new filter identifier.

3 Proposed "Filter identifier assignment" policy and process

To address the limitations of the current policy, and to better define the process of registering new filters and making them available to the HDF5 users' community, we propose the following new description in the HDF5 File Format Specification, the "Data Storage – Filter Pipeline" section:

"This [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. Filters supported by The HDF Group are documented below [the list of filters is available in the table]; information on the 3rd-party filters can be found at http://hdf5dev.pbworks.com/Community-Support-for-HDF5 (HDF5 Filters link).

To request a filter identifier please contact help@hdfgroup.org *with the following information*

1. *Contact information for developer requesting new identifier*
2. *Short description of the new filter*
3. *Links to any relevant information including licensing information*

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 the filters with identifiers from this range. "

4 Support for 3rd-party filters

This section describes the steps that The HDF Group will take in order to provide information about filters defined by 3rd-parties.

When a request for a 3rd-party HDF5 filter identifier is sent to the HDF HelpDesk, a designated person at The HDF Group

1. Assigns a unique identifier to that filter
2. Posts information provided by the submission on a Wiki page
3. Gives access to the Wiki page to the application developer

The HDF HelpDesk will refer users to that page when it receives any questions about the 3rd-party filters.

The designated person at The HDF Group will contact developers of the 3rd–party filters once a year to make sure that the filter is actively supported. If support for the 3rd –party filter cannot be confirmed, the corresponding Wiki Pages will be marked as "unsupported". Even in this case, the corresponding identifier will not be returned to the pool of available identifiers unless The HDF Group runs out of available values.

Acknowledgements

This work was supported by the GMQS project.

Revision History

April 8, 2009: Version 1 circulated for comment to Quincey.

April 9, 2009: Version 2 circulated for comment within The HDF Group.

June 25, 2009: Version 3 circulated for comment to the group, Francesc Alted and Andrew Collette.

July 19, 2009 Version 4 posted on Wiki at http://hdf5dev.pbworks.com/Community-Support-for-HDF5

Comments should be sent to epourmal@hdfgroup.org or koziol@hdfgroup.org

[References]

[1] HDF5 Reference Manual http://www.hdfgroup.org/HDF5/doc/RM/RM_H5Z.html

[2] HDF5 File Format Specification Version 2.0 http://www.hdfgroup.org/HDF5/doc/H5.format.html

Appendix: Examples of the documentation for bzip2, LZO and LZF filters that will be posted on Wiki. Information for each filter will be supported by a corresponding contact person.

Note: Wiki Page will need an additional disclaimer about the information posted there along with the links to the policy for requesting identifiers.

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:

Francesc Alted <faltet@pytables.org>

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:

Francesc Alted <faltet@pytables.org>

LZF Filter
Filter ID: 32000

Filter description:

This filter implements an alternative to DEFLATE for HDF5 datasets, using the free LZF library by Marc Alexander Lehmann.

The LZF filter provides high-speed compression with acceptable compression performance, resulting in much faster performance than DEFLATE, at the cost of a lower compression ratio. It's most appropriate for large datasets of low to moderate complexity, for which some compression is much better than none, but for which the speed of DEFLATE is unacceptable.

It's recommended to use the SHUFFLE filter with LZF, as it's inexpensive, supported by all current versions of HDF5, and can significantly improve the compression ratio.

Additional information:

The LZF filter is part of the h5py project, which implements a general-purpose interface to HDF5 from Python. A stand-alone C version of the filter will be publicly available as of h5py version 1.1, which may be freely used for any purpose under the terms of the h5py license (BSD). Maintenance of this filter is the responsibility of the author.

Links:

The h5py homepage is http://h5py.alfven.org
The LZF homepage is http://home.schmorp.de/marc/liblzf.html

Contact information:

Andrew Collette (UCLA)
Email: h5py at alfven dot org
Web: http://h5py.alfven.org