What’s new in HDF5 1.10

Gerd Heber

The HDF Group
1800 S. Oak Street, Suite 203
Champaign, IL 61820

http://www.hdfgroup.org/
Who is The HDF Group?

HDF Group has developed open source solutions for Big Data challenges for nearly 30 years.

Small company (~ 40 employees) with focus on High Performance Computing and Scientific Data

Offices in Champaign, IL + Boulder, CO

Our flagship platform – HDF5 – is at the heart of our open source ecosystem.

Tens of thousands use HDF5 every day, as well as build their own solutions (600 700 800+ projects on Github)

“De-facto standard for scientific computing” and integrated into every major analytics + visualization tool.
What does The HDF Group do?

**Products**
- HDF Capture: Software solution for PCAP Ingest + Storage (Beta)
- HDF5 Library
- Connectors: ODBC + Cloud (Beta)
- Add-Ons: compression + encryption

**Support**
- HDF Support Packages (Basic + Pro + Premier)
- Support for h5py + PyTables + pandas (NEW)
- Training

**Consulting**
- HDF: new functionality + performance tuning for specific platforms
- General HPC software engineering with fintech expertise (ex. MPI implementation for back testing)
- Metadata science and expert services
Overview
HDF5 from Release to Release
Understanding Compatibility and Versioning
Compatibility

Wikipedia:

• **Forward compatibility** is the ability of a system to accept input intended for later versions of itself.

• In technology, a product is said to be **backwards compatible** when it is able to take place of an older product, by interoperating with other products that were designed for the older product.
What (Gets Versioned) and Why

New versions of the library and file format because of:
• Bug fixes
• Performance improvements
• New features

All of the above may require:
• File format changes
• API changes
• New APIs
• Public structure changes
To Upgrade or Not to Upgrade

• Did they finally fix a bug I reported 5 years ago?
• Will I be able to read my old files with the new library?
• Do I need to re-link my application?
• Will MATLAB or IDL from 3 years ago work with files created by the new library?
• My colleagues and I use different versions of HDF5 libraries. Can we modify and access each others files?
HDF5 Library Versioning 1

HDF5 version number has a form of **X.Y.Z(-suffix)**

- X is called *major* version number
- Y is called *minor* version number (always even for public release)
- Z is called *release* number
- suffix is present in snapshots and release “candidates” (e.g. snap8, pre1)

**Examples:**

- **Releases:**
  - HDF5 1.6.5 and upcoming HDF5 1.8.0
  - Source tar file names hdf5-1.6.5.tar and hdf5-1.8.0.tar
- **Snapshots (source under development):**
  - HDF5 1.6.6-snap8 and HDF5 1.7.58
  - source tar file names hdf5-1.6.6-snap8.tar and hdf5-1.7.58.tar
HDF5 Library Versioning 2

HDF5 *release* number Z in a public release X.Y.Z

- No file format change
- No changes to the existing APIs
- No change to public data structures
- New APIs may be added by popular demand or by demand of the funding agencies (NASA, ASC) or as a result of a bug fix.
- Existing applications should be able to re-compile with the newest version
The HDF5 minor version number Y in a public release X.Y.Z is incremented each time when a new set of features is introduced:

• File format may change
• New APIs are added
• Old APIs may be removed or deprecated (will be removed in the next Y release)
• Public data structures may change (handled the same way as deprecated APIs)
HDF5 File Format Versioning 1

There is no HDF5 file format version number!

• Micro-versioning: each object and structure within an HDF5 file is versioned

• Updated “File Format Specification” is available with every public release

• There is no way to find what version of the library created or modified a particular file

Why did we choose such approach?
HDF5 File Format Versioning 2

Maximum file format compatibility principle

• By default the HDF5 files are written with the *earliest version* of file format that describes information, rather than always using the latest version possible.

• Assures best forward compatibility with the older versions (objects in new files can be read with old libraries if that object is “known” to the old libraries)
HDF5 Forward Compatibility

File Format
• Old versions of the library will read all objects in a file created by a newer library if objects are known to the old library
  • Example: 1.6.5 library will read a group in a file created by 1.8.0 version unless new 1.8.0 features are used (e.g. external links or compact groups)

Library APIs
• Application written to work with an older version will compile, link and run as expected with a newer version
  • May require configuration flag –enable–hdf5v1_Y to enable old APIs, data structures and behavior
HDF5 Backward Compatibility

File Format

• Newer version of the library will always read files created with an older version
  • Aside: HDF4 can read HDF4 files created in 1988 😊

Library APIs

• An application that doesn’t use new features will compile and link with the older library
Versioning References

• Release notes

• Backward and forward compatibility issues  
  https://support.hdfgroup.org/HDF5/faq/bkfwdc-
  compat.html

• API changes from release to release  
  http://hdfgroup.org/HDF5/doc_1.8pre/doc/ADGuid-
  e/Change

• File Format changes  
  http://hdfgroup.org/HDF5/doc/H5.format.html
New Features - HDF5 1.10
New Feature Overview

https://support.hdfgroup.org/HDF5/docNewFeatures/

New Features

• Single Writer Multiple Readers (SWMR) access
• Virtual Datasets (VDS)

Performance / Usability / Access Improvements

• Metadata Cache Image (MCI)
• Page Buffering and Paged Aggregation
• File Space Manager (FSM) enhancements
• Parallel HDF5 enhancements

Single Writer Multiple Readers (SWMR)

**What:** Read data from an HDF5 file that’s being written w/o a need for IPC between the writer process and one or more reading processes

**Use case:** Long running data acquisition, online analysis

**Caveats:**
- No Windows support (currently)
- Not supported in parallel HDF5 (currently)
Virtual Datasets (VDS)

**What:** Combine multiple (physical) datasets into a single (logical) dataset

**Use case:** High-speed cameras and detectors

**Caveats:**
- Not supported in parallel HDF5 (currently)
- File paths relative to executable (currently)
Metadata Cache Image (MCI)

**What:** Eliminate/reduce small I/O operations related to metadata, especially during file open/close.

**Use case:** (HPC) Systems with high latency / vertical data movement costs, e.g., parallel file systems; start with a *warm* cache, flush dirty entries in one go

**Caveats:**
- Not supported in parallel HDF5 (currently)
Page Aggregation and Buffering

**What:** Optimize file space allocation for small pieces of meta- and raw-data through (fixed-size, aligned) paging + eliminate small and random I/O accesses

**Use case:** Accelerate metadata and small raw-data I/O ops on parallel file systems (combat high latency)

**Caveats:**
- Intended to be used together
- Not compatible with the multi/split-VFDs (currently)
Parallel HDF5 Enhancements

**What:** Reduce metadata I/O overhead through collective operations (prevent read and write storms)

**Use case:** Metadata-rich HDF5 profiles, e.g., CGNS

**Caveats:**
- Your mileage may vary
Making HDF5 1.10 files compatible with 1.8

<table>
<thead>
<tr>
<th>If you use the following feature in 1.10</th>
<th>Use this tool to convert to 1.8</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>H5Pset_libver_bounds(..., H5F_LIBVER_LATEST, H5F_LIBVER_LATEST)</td>
<td>h5repack</td>
<td>All data in the file has to be repacked.</td>
</tr>
<tr>
<td>SWMR</td>
<td>h5format_convert</td>
<td>The tools rewrites the file’s metadata only in 1.8 format; no raw data is rewritten.</td>
</tr>
<tr>
<td></td>
<td>h5repack</td>
<td>Use h5repack if no modification to the original file is desired.</td>
</tr>
<tr>
<td>Partial Edge Chunks</td>
<td>h5repack</td>
<td>Rewrites the data using 1.8 format for chunked datasets.</td>
</tr>
<tr>
<td>VDS</td>
<td>h5repack –l &lt;layout&gt;</td>
<td>Repacks VDS according to the &lt;layout&gt; storage option.</td>
</tr>
<tr>
<td>Metadata cache image</td>
<td>h5clear –m</td>
<td>Removes cache image from the file making it compatible with the 1.8 format.</td>
</tr>
</tbody>
</table>

1 File format that can be read by the HDF5 library 1.8
References

• The HDF Group [https://www.hdfgroup.org/](https://www.hdfgroup.org/)
• HDF5 1.8.x [https://support.hdfgroup.org/HDF5/release/obtain518.html](https://support.hdfgroup.org/HDF5/release/obtain518.html)
• HDF5 1.10.x [https://support.hdfgroup.org/HDF5/release/obtain5.html](https://support.hdfgroup.org/HDF5/release/obtain5.html)
• New features overview and docs [https://support.hdfgroup.org/HDF5/docNewFeatures/](https://support.hdfgroup.org/HDF5/docNewFeatures/)