Supporting HDF5 1.8 in HDF5 Command Line Tools

Peter Cao
xcao@hdfgroup.org
The HDF Group

HDF5 1.8 includes a number of new features and file format changes that offer users of HDF5 substantial performance improvements and expanded capabilities. These features can only be accessed via new API calls in users' applications. Support for some of the features has been added to the HDF5 command line tools since the first release of the HDF5 1.8 library. However, the tools have not been kept up to date with respect to all the features introduced by the HDF5 1.8 library. This document discusses the work needed to support the remaining HDF5 1.8 features in the tools.

The purpose of the document is to identify the work needed to support the HDF5 1.8 changes in the HDF5 tools and the priorities of those tasks. Once we have identified the tasks and prioritized them, we will need an implementation plan or design for each specific task, for example supporting external links.

Table of Contents

1 Introduction

HDF5 version 1.8 represented a major update to the HDF5 library and file format from HDF5 version 1.6. The changes provided new capabilities and improved performance. While users can take advantage of the new features by using new API calls in the library, the HDF5 command line tools have not been updated to meet users' needs for handling the features introduced in HDF5 1.8.

Since the first release of HDF5 1.8 (version 1.8.0) in February 2008, HDF5 1.8 has been adopted by many users. Consequently, there is great demand for supporting the HDF5 1.8 features in the HDF5 command line tools. However, the HDF5 tools are not keeping up with the new changes to the HDF5 library.

This document addresses the current issues in HDF5 tools and provides guidance on how to support the HDF5 1.8 features in the tools. The document is organized in as follow:

- An introduction to the major changes in HDF5 1.8
- A brief description of the current tools
- Recommended improvements for the tools
- Task list and priorities

2 Major changes in HDF5 1.8

HDF5 1.8 introduced some major changes in the library functions and file format. The following section is an introduction to these changes. For details, please visit "What's New in HDF5 1.8.0" at http://www.hdfgroup.org/HDF5/doc/ADGuide/WhatsNew180.html.

2.1 Change in file format

Currently two versions of the HDF5 file format specification are defined. Version 1.1 (or the older version) is used in HDF5 1.6 and version 2.0 (the newer version) can be used in HDF5 1.8. Files and objects created with the newer format cannot be accessed through the HDF5 1.6 library. The HDF5 development team has provided the following compatibility capabilities:

- ***Limited forward compatibility***: HDF5 1.6 is able to read objects that are known to the HDF5 1.6 library in a file created by HDF5 1.8. However, HDF5 1.6 will not be able to recognize files or objects created according to the new file format specification (version 2.0).
- ***Full backward compatibility***: HDF5 1.8 is able to read all files and objects created by HDF5 1.6.

To address compatibility concerns, the HDF5 library is designed to provide the greatest possible format compatibility, writing the oldest version of the format that can be used to store an object. By default, only when an application requests a particular feature that requires a newer format version will the HDF5 library create objects using later format versions. If a new version of all objects is needed, or if all objects should be prohibited from using the new format version, applications can request format version limits with the H5Pset_libver_bounds() API routine. As a result, there can be four types of files:

A) A file and all objects in the file are created with the older version.

B) A file is created with the older version but some objects are created with the newer version.

C) A file is created with the newer version but some objects are created with the older version.

D) A file and all objects in the file are created with the newer version.

An application built with HDF5 1.8 is able to access all four types of files. If an application is built with an older version of the HDF5 library, it will be able to completely access type A) files. However, it may have problems with the other types of files:

- ***Type B):*** the application will be able to open the file but will not be able to access the objects created with the newer version. It may not even know of the existence of those objects. For example, when John sends a file to Jane and asks Jane to look at certain objects in the file. Jane may not be able to see those objects if she is using an older version of the library.
- ***Types C) and D):*** the application will not be able to open the file at all, even though some objects were created with an older version as in a type C) file. Users may have no idea whether the file is corrupted or created in a newer format.

2.2 New features

2.2.1 Compact groups

In addition to the *indexed link storage* for groups used in HDF5 1.6, HDF5 1.8 adds a new method of storing links for groups referred to as *compact link storage*. Compact link storage allows groups containing only a few links to take up much less space in the file. In contrast, indexed link storage for groups provides a faster and more scalable method for storing and working with groups containing many links.

A group stored in the "new" format can transition between compact link storage and indexed link storage based on the number of links in the group. The threshold for switching between the two storage formats is configurable by using the function H5Pset_link_phase_change().

If the group is stored with the new group format, both the compact and indexed forms of the group will not be readable by older versions of the library. The old group format used v1 B-trees and local heaps and the new group format's indexed storage uses v2 B-trees and fractal heaps.

2.2.2 External links

An external link in a group references an object in a different HDF5 file. External links are new in HDF5 1.8. Applications built on HDF5 1.6 will not be able to access objects pointed by external links. Also, creating an external link in a group that is stored in the "old" group format will cause the group to be converted into the "new" group format.

External links are heavily used by some of our major customers. We have recently added support for this feature to many of the tools, including h5dump, h5diff, h5ls, h5copy and h5diff.

2.2.3 Link creation order

HDF5 1.8 supports tracking, indexing, and iterating over links in groups by creation order. Like external links, requesting link creation order support when creating a group will cause that group to be stored in the "new" group format. Support for showing link creation order has been added to some of the tools such as h5dump, but a consistent design among all tools is needed.

2.2.4 Shared object header messages (SOHM)

To conserve space in an HDF5 file, large header messages that are used repeatedly in the file can be designated as shared.

2.2.5 UTF-8 Unicode encoding

UTF-8 Unicode encoding is supported in HDF5 1.8 for string datatypes in datasets, the names of links, and the names of attributes. Currently HDF5 tools do not support UTF-8 unicode strings.

2.2.6 Create intermediate groups

Intermediate groups that do not yet exist can be created when creating or copying an object in a file.

2.2.7 Object copying

HDF5 1.8 allows copying an HDF5 object to a new location within a file or in a different file..

2.2.8 Conversion between datatype and text

This feature enables the creation of a datatype from a text definition of that datatype and vice versa.

3 New features for the command line tools

3.1 Current tools

The HDF5 command line tools fall into five categories: view, analyze, organize, import, and export. For detailed descriptions of the tools, please visit "Software Using HDF5" at http://www.hdfgroup.org/products/hdf5_tools/. The following is a list of the HDF5 command line tools.

- ***View***: h5dump, h5ls, and h5watch
- ***Analyze***: h5diff, h5stat, h5perf, h5check, and h5debug
- ***Organize***: h5repack, h5copy, h5repart, and h5mkgrp
- ***Import***: h5import, gif2h5
- ***Export***: h5dump (h5dump can serve as an exporter by dumping data to ASCII, binary, and xml)

Various changes in HDF5 1.8 affect different tools. Some features have already added to some of the tools. Details are discussed below.

3.2 Compact groups

The following option can be added to the tools to deal with compact groups:

- Allow users to set threshold of compact groups

Tools related to this feature include h5repack, h5copy, and h5mkgrp.

3.3 External and soft links

Tool features related external and soft links include:

- ***An option to follow links***. This option has been added to h5ls, h5diff, and h5copy. Other tools related to this feature include h5repack and h5dump. We should revise the current implementation so that the option and behavior will be consistent among all the tools.
- ***A tool to create, modify, and delete links***: we currently do not have a tool that can create, modify, or delete links. We recommend creating a new tool, h5ln (similar to 'ln' command in unix) or h5link (which is more HDF5 specific), for creating all types of links.

3.4 Link creation order

Add options to show objects in creation order and to enable creation order. Related tools may include:

- ***View***: h5dump, h5ls
- ***Analyze***: h5diff
- ***Organize***: h5repack, h5copy, and h5mkgrp

3.5 Shared object header messages (SOHM)

Add an option to h5repack to set what messages can be shared.

3.6 UTF-8 Unicode encoding

UTF-8 Unicode encoding is supported for string datatypes in datasets, names of links, and names of attributes in HDF5 1.8. This feature needs to be added to the following tools:

- ***View***: h5dump, h5ls and h5watch
- ***Analyze***: h5diff and h5debug
- ***Organize***: h5repack, h5copy, and h5mkgrp

3.7 Create intermediate groups

Creating intermediate groups was implemented in h5mkgrp. This feature needs to be added to h5copy.

3.8 Object copying

The copy tool, h5copy, has been added to HDF5 1.8 releases. The tool allows various options when an object is copied from one location to another (within a file or cross files). We need to revise the current the implementation to make sure the options and behavior are consistent with other tools.

3.9 Conversion between datatype and text

Two options can be added to the current tools:

- An option to write a datatype to DDL text with h5dump and h5ls.
- An option to create a datatype/dataset from DDL text in h5import.

4 Compatibility

We currently do not have any tool that handles issues of file format compatibility. Features such as showing format version information for a file's superblock and the objects in the file and converting files between an older and newer version are very important to users.

To address file format compatibility issues, we propose the following two approaches for users:

- ***Retrieving version information for a file's superblock and for objects in a file***: when an application fails to open a file or object, it would be very useful to know if the file is corrupted or just incompatible with the version of the library used by the application. Solutions include implementing a new tool for this purpose or adding a new feature to h5check to check the version information.
- ***Re-writing an existing file according to a specific version of the library (upgrading/downgrading)***: an option can be added to h5repack or h5copy that allows users to re-write the whole file or specific objects in a file to a specific version.

5 General design issues and recommendations for improvement

5.1 Major issues in the current tools

Since the first release of HDF5 1.8, users have requested support for HDF5 1.8 library features in the tools. Some features have been added to the tools based on those requests. For example, a new tool, h5copy, was developed based on the new function H5Ocopy(). However, there are still some major issues regarding the general design.

- ***Lack of design and planning***: some HDF5 1.8 features have been added to the tools for a particular purpose or request. However, the implementation is lacking a coherent design and plan, which causes inconsistencies among the tools' interfaces and feature sets. Careful design and planning will bring better results. For example, designing common functions for all tools will result in less duplication and more efficient code.
- ***Inconsistent behavior and user interface***: features added to tools are not consistent across all the tools. For example, the options and behaviors for dealing with external links are different among the tools. Inconsistency creates confusion for users and adds more work when developing and maintaining the tools.

5.2 Recommendations for improvement

We recommend the following improvements for the tools when a new feature or tool is added to support the changes added to HDF5 1.8.

5.2.1 Development procedure

To implement a new feature or a new tool, we need to follow our development procedures. Development documents should include requirements document and an RFC or design document.

5.2.2 User interface

Using the same user option (or flag) for the same feature among the tools will avoid confusion for users. Solutions include:

- ***Using long option names***: Some tools, such as h5dump, have run out of reasonable single letter options. Also, in some cases the same letter (option) has different meanings in different tools. For example, "-l" option is used to specify dataset layout in h5repack. The same option is used for displaying members of a compound datatype in h5ls. But h5dump uses "-l" for following soft links. Transitioning to using long options for all tools, e.g. "—follow-links" for following symbolic links (soft links and external links), is a path out this problem.
- ***Deprecating old options***: as we replace the old options with the long options, we will still keep the old options, but document them as deprecated, so that the changes will not break users' scripts.

5.2.3 Common functions

We will develop a set of general functions in a tools library. Using general functions in tools for dealing with HDF5 1.8 features will avoid repeated code. General functions will help us have better code and less maintenance and consistent behavior among the tools

5.2.4 Testing

Our current tool testing is not efficient. Whenever we fix a bug or add features to tools, we have to write code to generate test files. The process is very time consuming and error-prone. A better approach is to have a standard set of test files or programs that generate test files that all tools can use.

Using common test files and testing code will also reduce the amount of development work and lead to better product quality.

6 Task list with priorities

The tasks are listed in the order of priorities in the table below. The priorities are defined as

- ***1 for high priority***: Requested by major customers.
- ***2 for medium priority***: Important to general users but not urgent
- ***3 for low priority***: Not important to general users.

The priorities are given based on the features. Each tool may have different priories. For example, following external/soft links has the highest priority; however, it is not important to h5stat. The priorities will be fine-tuned at the design and implementation of each feature. One of main goals here is to identify what features to work first.

Table 1 -- Task list with priorities

Task#	Task	Related tools	Status	Priority
A	**External/soft links**			

A.1	Follow links	**View:** h5dump, h5ls **Organize**: h5repack, h5copy **Analyze**: h5diff, h5stat	Partially done	1
A.2	Create/modify/delete links	**Organize**: h5link (new)	No work	2
B	**Object copying**	**Organize**: h5copy	Mostly done. Need to revise options	1
C	**Create intermediate groups**	**Organize**: h5copy, h5mkgrp, h5link (new)	Done for h5mkgrp. Need work on h5copy	1
D	**Compatibility**			
D.1	Display version information	**View:** h5dump, h5ls **Analyze**: h5diff, h5stat, h5check	No work	2
D.2	Re-writing an existing file to a specific version	**Organize**: h5repack, h5copy	No work	2
E	**Compact groups**			
	Allow users to set threshold	**Organize**: h5repack, h5copy, h5mkgrp	No work	2
F	**Link Creation order**			
F.1	Show objects in creation order	**View:** h5dump, h5ls **Analyze**: h5diff, h5stat, h5debug	Done for h5dump No work for other tools	2
F.2	Enable creation order	**Organize**: h5repack, h5copy, h5mkgrp	No work	2
H	**UTF-8 Unicode encoding**	**View:** h5dump, h5ls **Organize**: h5copy, h5mkgrp **Analyze**: h5diff, h5debug	No work	3
I	**Conversion between datatype and text**			3
I.1	Write datatype to DDL text	**View**: h5dump, h5ls	No work	3
I.2	Create datatype/dataset with DDL text	**Import**: h5import	No work	3

Revision History

March 30, 2010: Version 1 draft for Quincey and Elena to review.

April 9, 2010: Version 2 draft for circulating internally. Updated based on inputs from Quincey, Elena inputs, and Ruth