Support HDF5 1.8 in the HDF-Java Object Layer

Peter Cao
xcao@hdfgroup.org
The HDF Group

HDF5 1.8 includes a number of new features that offer many users of HDF5 substantial performance improvements and expanded capabilities. Many of these features can only be accessed via revised API calls. To support these new APIs and new features in the HDF-Java products, we need make changes to the current the HDF-Java products. The changes may affect applications that use the HDF-Java products, especially the HDF5 Java wrapper.

The work for supporting HDF5 1.8 APIs in JHI5 is in progress and is expected to be completed in April 2010. For details of the JHI5 work, see RFC at https://www.hdfgroup.uiuc.edu/RFC/HDF5/HDF-Java/RFC_Support_HDF518_in_Java.pdf.

This document discusses what work is needed to support HDF5 1.8 in the HDF-Java object layer. The work will be carried out right after the completion of JHI5 work.

The purpose of this document is to scope out the work needed to be done and look for feedback on technical issues. Please send your concerns and suggestions to Peter Cao at xcao@hdfgroup.org.

Table of Contents

1 Introduction

HDF5 1.8 represents a major update to the HDF5 Library, utilities, and file format. The changes introduced in HDF5 1.8 provide new capabilities and improve performance.

There are three distinct HDF5 Java Products:

- Java HDF5 Interface (JHI5): the Java Native Interface to the standard HDF5 library.
- Java HDF Object Package: a Java package that implements HDF data objects in an object-oriented form.
- HDFView: a visual tool for browsing and editing HDF4 and HDF5 files.

HDFView is built on the Java HDF Object Package, which in turn is built on the Java HDF5 Interface. Modifications will be needed in all three products to support HDF5 1.8 features. The necessary changes for JHI5 will be completed in April 2010. Supporting HDF5 1.8 in HDFView will be addressed in the future. This RFC will mainly focus on the update of HDF-Java object layer.

2 Changes to the HDF-Java object package (layer)

This section describes what changes are needed in the HDF-Java object package in order to support HDF5 1.8 functions.

HDF-Java object package simplifies accesses to HDF files by encapsulating the details of calls to the HDF library in higher-level classes and methods. For example, the image object provides a method to read an image from an HDF5 file. This method may invoke more than ten calls to the HDF5 library to accomplish this task. For this reason, many features in HDF5 are not supported by the object layer. For advanced features, users have to use the JHI5. For the same reason, many new features introduced by HDF5 1.8 will not be included in the HDF-Java object layer.

2.1 Introduction to the HDF-Java object package

The HDF Object Package is a Java package that provides an object-oriented interface to HDF data objects. The HDF Object Package, ncsa.hdf.object, provides classes that reflect the fundamental objects of HDF4 and HDF5. For more information on HDF-Java object package, visit http://www.hdfgroup.org/HDF-Java -html/hdf-object/index.html. This work will focus on HDF5 object package.

Implementing classes of ncsa.hdf.object for HDF5 are defined in the object package, ncsa.hdf.object.h5 package. The HDF5 object package contains five basic classes:

- **H5File** -- H5File is an implementation of the FileFormat class for HDF5 files.
- **H5Group** -- An H5Group object represents an existing HDF5 group in file.
- **H5Datatype** -- This class defines HDF5 data type characteristics and APIs for a data type.
- **H5ScalarDS** -- H5ScalarDS describes a multi-dimension array of HDF5 scalar or atomic data types, such as byte, integer, short, long, float, double and string, and operations performed on the scalar dataset.

- **H5CompoundDS** -- The H5CompoundDS class defines an HDF5 dataset of compound data types.

For more information on the HDF5 object package, please visit http://www.hdfgroup.org/HDF-Java - html/javadocs/.

2.2 Types of changes to the HDF-Java object package

There will be two types of changes to the HDF-Java object package:

- Changes that do not affect current users -- no change are needed to the current applications that use HDF-Java object package.
- New features/functions – changes are needed only if users want to use new the features.

2.3 Changes that do not affect current users

These changes will not affect current applications that use HDF-Java object package. The changes are transparent to users.

2.3.1 Deprecated methods

There are 40 deprecated functions (see appendix A). Most of the functions except functions in H5E and H5P packages are used in the HDF-Java object package. These deprecated functions will be replaced with the new ones.

2.3.2 Performance improvement

Some of the new functions in HDF5 1.8 can increase the performance of HDF-Java . Based on the first evaluation, following function(s) will be used to replace the current Java implementation for better performance:

- H5Ocopy() -- this function in the HDF5 library has better performance than the current Java copy() function.

2.4 New features/functions

2.4.1 General features

HDF5 1.8 allows users to create object with a specified version. By default, objects will be created the earliest possible format that will handle the data being stored and accommodate the action being taken. Changes will be made to HDF-Java object to support this feature.

Changes:

- H5File – new method will be added to allow users to set the bounds of library versions, probably overloading current methods by adding extra parameters.

2.4.2 Groups

Compact-or-indexed groups enable much-compressed link storage for groups with very few members and improved efficiency and performance for groups with very large numbers of members.

H5Pset_link_creation_order sets creation order tracking and indexing for links in a group.

We will add new options to allow users to set link storage and to use creation order other than alphabetical order.

Changes:

- H5File – pass user options for link storage and creation order to H5Group.
- H5group – Add overloading functions to take user options for link storage and creation order.

2.4.3 External inks

External links enable the insertion of a link into an HDF5 group in one file that points to an HDF5 object in a different HDF5 file.

With the completion of the new HDF-Java JNI, reading/writing content of an object pointed by an external link will work with the HDF-Java object package. However, the current object package does not allow users to create an external link and get link information.

Changes:

- H5Group – add new method to allow users to create an external link.
- All objects – retrieve link information (external file and path).

2.4.4 Attributes

Like groups, we will allow users to retrieve attributes in creation order. By default, attributes are listed in alphabetical order.

Changes:

- All objects – pass user option for creation order.

3 Compatibility issues

Backward compatibility relates to the ability of a newer HDF5 Library to read files created by an older HDF5 Library. HDF5 Library is always backwardly compatible. Forward compatibility relates to the ability of an older HDF5 Library to read files created by a newer HDF5 Library. An older library should be able to read objects that are known to that library. For more information on HDF5 backward/forward compatibility, visit,

http://www.hdfgroup.org/HDF5/doc/ADGuide/CompatFormat180.html.

HDF-Java objects deal with compatibility issues according to the library:

- File level – for any compatible files, H5File will open as valid HDF5 files. For a file that is not compatible to the library version used by HDF-Java, H5File will not open the file.
- Object level – for any compatible object (Group or Dataset), HDF-Java object will recognize it as a valid object. For an object that is not compatible to the library that is used by the hdf-java object, it will not be recognized.

4 Prioritized tasks

The following is a list of tasks discussed above with priorities, 1 – high, 2 – medium, 3 – low. The work will include implementation, documentation, and testing. The estimation of number of hours is based on the fact that the staff will have good understanding of the hdf-java products.

Task	Priority	Notes
Replace deprecated methods	1	
Use H5Ocopy()	1	
Set the bounds of library versions	1	
Create external links	1	
Retrieve link information	2	
Use link creation order	2	
Set link storage	2	
Use creation order in attributes	3	
Other	3	
Total		

Revision History

February 17, 2010:	Version 1 general issues and outline.
March 15, 2010:	Version 2 updated based on the feedback from circulating internally.

5 Appendix A: A list of deprecated functions in HDF5 1.8

Deprecated HDF5 functions	Used in HDF-Object layer
H5Acreate	Yes
H5Aopen_name	Yes
H5Aopen_idx	Yes
H5Aget_num_attrs	Yes
H5Aiterate	Yes
H5Dcreate	Yes
H5Dopen	Yes
H5Dextend	No
H5Eclear	No
H5Eget_auto	No
H5Epush	No
H5Eprint	No
H5Eset_auto	No
H5Ewalk	No
H5Eget_major	No
H5Eget_minor	No
H5Fget_info	Yes
H5Gcreate	Yes
H5Gopen	Yes
H5Glink	Yes
H5Gmove	Yes
H5Gunlink	Yes
H5Gget_linkval	Yes
H5Gset_comment	Yes
H5Gget_comment	Yes
H5Giterate	Yes
H5Gget_num_objs	Yes
H5Gget_objinfo	Yes
H5Gget_objname_by_idx	Yes
H5Gget_objtype_by_idx	Yes
H5Pregister	No
H5Pinsert	No
H5Pget_filter	No
H5Pget_filter_by_id	No
H5Pget_version	No
H5Rget_obj_type	No
H5Tcommit	Yes
H5Topen	Yes
H5Tarray_create	Yes
H5Tget_array_dims	Yes