RFC: h5diff – Exclude Object(s) from Comparison

Jonathan Kim (jkm@hdfgroup.org)

The h5diff command-line utility compares objects in HDF5 files and reports differences. Currently, h5diff does either pairwise comparison of all objects in the files or comparison of two particular objects.

This RFC proposes adding an option to h5diff that allows the user to exclude object(s) from the pairwise comparison.

1 Introduction

The h5diff[1] command-line utility has ability to compare entire HDF5 files or specific objects (groups, datasets, named datatypes, symbolic links).

When comparing HDF5 files, h5diff compares objects with matching paths in the two files. This is referred to as pairwise comparison.

This RFC proposes a new option that will cause h5diff to exclude specified object(s) from the pairwise comparison when comparing HDF5 files.

2 Motivation

In some circumstances, h5diff users may want to compare all objects in two HDF5 files, with the exception of a small number of objects that they know exist in only one file, or that they know to be different. Two examples demonstrate when excluding an object from comparison would be useful.

2.1 Example 1

File1.h5 has these objects: /g1, /g1/d1, /g1/d2, /d1, /g2, /g2/s1, /g2/d1

File2.h5 has these objects: /g1, /g1/d1, /g1/d2, /d1, /g2, /g2/s1

The command "h5diff File1.h5 File2.h5" will return an exit code of "1", indicating the two files are different, because File2.h5 does not have the object /g2/d1. However, the output from h5diff, when run without the –v option, will not show any differences.

The user may be aware of the extra object (/g2/d1) in File1.h5 and want to exclude it from comparison, having h5diff compare the remaining objects in the files, and returning an exit code based only on those objects.

[1] Refer to the h5diff reference manual entry for a more extensive discussion of h5diff's behavior.

2.2 Example 2

File3.h5 has these objects: /G1, /G1/D1, /G1/D2, /D1, /G2, /G2/S1, /G2/D1, /D_timestamp

File4.h5 has these objects: /G1, /G1/D1, /G1/D2, /D1, /G2, /G2/S1, /G2/D1, /D_timestamp

The user may not care if the D_timestamp objects are different, but may want to compare all other objects in the files.

2.3 Current Approach

Currently, there is no easy way to exclude object(s) from comparison. Instead, the user must run h5diff multiple times, compare a specific pair of objects with each run, and report any differences.

This approach could be implemented as follows:

1. Generate a list of absolute paths for all objects in each file.
2. Remove duplicate entries from the list.
3. Remove the path(s) of the object(s) that should not be compared.
4. Use a script to run "h5diff *file1 file2 object"* multiple times, once for each object that remains in the pruned object list.

While this approach is possible, it becomes unwieldy as the number of objects in the file increases. Furthermore, executing h5diff multiple times introduces a performance penalty.

3 Proposed Solution

This RFC proposes a new option that will cause h5diff to exclude specified object(s) from the pairwise comparison when comparing HDF5 files. This solution will allow the user in the examples presented above to perform the desired comparisons without the added effort required by the current approach.

3.1 --exclude Option

We propose calling the new option *--exclude,* with the following usage:

h5diff ***--exclude*** *exclude_object file1 file2*

The argument following *--exclude,* denoted by *exclude_object* in the sample command line, specifies the object that will be excluded from the pairwise comparison. The excluded object can be a group, dataset, named datatype, or symbolic link (soft link or external link), and must be expressed as an absolute path from the root group.

With the exclude option, all occurrences of the specified object are excluded from comparison, regardless of whether the object occurs in file1, in file2, or in both files.

If the excluded object is a group, the group and all objects in the hierarchy below the group are excluded from the pairwise comparison.

The exit code and output from h5diff will be based on pairwise comparison of the objects in file1 and file2 that have not been excluded.

3.2 Excluding Multiple Objects

If multiple objects are to be excluded, the *--exclude* option must be repeated for each excluded object:

h5diff ***--exclude*** "/g1/d2" ***--exclude*** "/g2/d1" … *file1 file2*

While repeating the *–exclude* option may seem cumbersome, it will simplify the construction of command lines for automated scripting. Furthermore, since HDF5 allows all characters to be used when naming an object (there are no reserved or special characters), separating absolute pathnames (objects) specified in a single quoted string would not be straightforward.

4 Use Cases

4.1 Case 1: Excluding an extra object

Consider the motivating Example 1 in Section 2.1, where

File1.h5 has these objects: /g1, /g1/d1, /g1/d2, /d1, /g2, /g2/s1, /g2/d1

File2.h5 has these objects: /g1, /g1/d1, /g1/d2, /d1, /g2, /g2/s1

To exclude the extra object /g2/d1 that only appears in File1.h5 from comparison, the command would be:

```
>> h5diff –exclude "/g2/d1" File1.h5 File2.h5
```

The h5diff exit code will be 0 if the pairwise comparison of other objects in the file found no differences, and 1 if some differences were found in those objects.

4.2 Case2: Excluding an object that exists in both files

Consider the motivating Example 2 in Section 2.2, where

File3.h5 has these objects: /G1, /G1/D1, /G1/D2, /D1, /G2, /G2/S1, /G2/D1, /D_timestamp

File4.h5 has these objects: /G1, /G1/D1, /G1/D2, /D1, /G2, /G2/S1, /G2/D1, /D_timestamp

To exclude the object /D2_timestamp (that is expected to be different) from comparison, the command would be:

```
>> h5diff –exclude "/D_timestamp" File3.h5 File4.h5
```

The h5diff exit code will be 0 if the pairwise comparison of other objects in the file found no differences, and 1 if some differences were found in those objects.

4.3 Case3: Excluding objects that exist in only one file; excluding group objects

A user has two HDF5 files, with the objects shown:

File5.h5 has these objects: /gg1, /gg1/dd1, /gg1/dd2, /dd1

File6.h5 has these objects: /dd1, /gg2, /gg2/ss1, /gg2/dd1

Based on the path names, and the fact that they have objects under them, we can infer that /gg1 and /gg2 are group objects.

The h5diff command

```
>> h5diff –exclude “/gg1” –exclude “/gg2” File5.h5 File6.h5
```

excludes the group object /gg1 (and all objects under it) and group object /gg2 (and all objects under it) from comparison. /gg1 only exists in File5.h5 and /gg2 only exists in File6.h5.

This command causes h5diff to do a comparison on the /dd1 objects in File5.h5 and File6.h5. If the /dd1 objects are the same, the h5diff exit code will be 0. If differences are found, it will be 1.

This command, with the example files shown, is equivalent to:

```
>> h5diff File5.h5 File6.h5 “/dd1”
```

5 Additional Features for Future Consideration

These feature requests came up in the process of discussing the RFC. They will not be implemented in the initial release, but are recorded for future reference and possible implementation at a later date.

- ‘*—exclude-all-paths-object*’: If the specified object is accessible via multiple paths (has multiple hard links to it), exclude all of those paths from comparison. If this option is implemented, it will also be necessary to clarify how soft links to the object will be treated (i.e., will they also be excluded), and if the treatment of soft links to the object will depend on whether or not --follow-symlinks option is specified.
- Allow the use of a configuration file to specify multiple objects to exclude. This will be especially helpful when the object paths are too long to fit on one command line.
- Support multiple paths with one invocation of the *–exclude* option. Before this feature can be added, a strategy for reserving and/or escaping special characters in object paths must be developed.
- Allow wild cards in object path names. Before this feature can be added, a strategy for reserving and/or escaping special characters in object paths must be developed.
- Add an option, such as “--common-objects-only”, to specify that h5diff should only perform pairwise comparisons on objects that are common to both files. With this option, a user could more easily indicate that they don’t want the occurrence of an object in one file but not the other to trigger a “files are different” condition.

  In particular, the Use Case 1 command:

  ```
  >> h5diff --exclude “/g2/d1” File1.h5 File2.h5
  ```

  could be replaced by:

  ```
  >> h5diff --common-objects-only File1.h5 File2.h5
  ```

  And, the Use Case 3 command:

  ```
  >> h5diff --exclude “/gg1” --exclude “/gg2” File5.h5 File6.h5
  ```

  could be replaced by:

  ```
  >> h5diff --common-objects-only File5.h5 File6.h5
  ```

For users who frequently want to compare only objects that are common to both files, this option would allow them to do so easily, and would not require that the enumerate (or even know) the objects that are in only one of the two files.

If the implementation of the --exclude option makes the --common-objects-only option very straightforward, it may be included in the initial release.

Acknowledgements

This work was prompted by a request from Mark Linda, of NASA, who frequently wants to compare some, but not all, objects in HDF5 files.

Ruth Aydt (aydt@hdfgroup.org) provided editorial support.

Revision History

May 13, 2010:	Version 1 circulated for comment within The HDF Group.
May 17, 2010:	Version 1.2 updated from the internal meeting and feedbacks within the HDF Group.
May 24, 2010:	Version 1.3 changed option from --ignore-path to --exclude to be more descriptive and consistent with existing h5diff reference manual page. Removed discussion of hard and soft links, which were orthogonal to the main purpose of the RFC. Made other editorial changes. Expanded Additional Features discussion.