RFC: h5diff – Exclude Object(s) from Comparison

Jonathan Kim ([jkm@hdfgroup.org](mailto: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. It also proposes approaches for excluding attribute(s) from comparison.

# Introduction

The h5diff[[1]](#footnote-1) command-line utility has the ability to compare entire HDF5 files or specific objects (groups, datasets, committed 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 object(s) from the pairwise comparison by specifying path(s) that should not be traversed.[[2]](#footnote-2)

The RFC also proposes approaches for excluding attribute(s) from comparison.

# Motivation

h5diff is often run as part of a scripted batch process that relies on h5diff’s exit code to determine the results of the comparison, rather than on a visual scan of the text output. In some circumstances users may want to compare most of the objects in two HDF5 files, excluding 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 and basing the exit code and text output on the comparison of the non-excluded objects would be useful.

## 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.

## 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 and return an exit code based on the comparison of those objects.

## Current Approach

Currently, there is no easy way to exclude object(s) from comparison.

One approach would be to 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.

Another approach is to run “h5diff -v *file1 file2* > diffout*”*, then use a series of post-processing commands (grep, sed, awk, etc.) to filter out the known differences from the “diffout” text file. This approach can be quite slow, and is problematic when the size of “diffout” is large.

# Proposed Solution

This RFC proposes a new option that will cause h5diff to exclude object(s) from the pairwise comparison by specifying path(s) that should not be traversed. 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.

## --exclude-path Option

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

h5diff ***--exclude-path “****path” file1 file2*

The argument following *--exclude-path,* denoted by “*path”* in the sample command line, specifies a path that h5diff should not traverse during the pairwise comparison. The specified path may exist in file1, file2, or both files. The object identified by the path may be a group, dataset, committed datatype, or symbolic link (soft link or external link).

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.

## Excluding Multiple Objects

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

h5diff  ***--exclude-object*** “/g1/d2”   ***--exclude-object*** “/g2/d1”  … *file1 file2*

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

# Use Cases

## 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-object “/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 any differences were found in those objects.

## Case 2: 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 /D\_timestamp (that is expected to be different) from comparison, the command would be:

>> h5diff --exclude-object “/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 any differences were found in those objects.

## Case 3: 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-object “/gg1” --exclude-object “/gg2” File5.h5 File6.h5

excludes the group object /gg1 (and all objects under it) and the 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 of 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”

# Additional Features Currently being Discussed

The features described in this section are currently being discussed and The HDF Group is actively seeking feedback regarding the proposed approaches.

## Excluding Attribute(s)

Users have expressed interest in excluding attributes from comparison. There are four use cases that may be implemented at some point, and that the design should support. These are assigned numbers that begin with “A”, to distinguish them from the use cases for excluding objects discussed previously:

* Case A1: For a specific object, exclude a specific attribute
* Case A2: For a specific object, exclude all attributes
* Case A3: For all objects, exclude a specific attribute
* Case A4: For all objects, exclude all attributes

While all four use cases may not be supported by h5diff initially, the design should accommodate them all in a straightforward manner.

### Proposed approaches

Two approaches for specifying the attribute(s) to exclude from comparison are currently being considered.

#### Approach One

Use a single option, *--exclude-object-attribute*, and a wildcard argument, “\*”, as shown:

* Case A1: For a specific object, exclude a specific attribute

h5diff ***--exclude-object-attribute*** “*full\_path\_to\_object*” “*attribute\_name*” *file1 file2*

* Case A2: For a specific object, exclude all attributes

h5diff ***--exclude-object-attribute*** “*full\_path\_to\_object*” “*\**” *file1 file2*

* Case A3: For all objects, exclude a specific attribute

h5diff ***--exclude-object-attribute*** “\*” “*attribute\_name*” *file1 file2*

* Case A4: For all objects, exclude all attributes

h5diff ***--exclude-object-attribute*** “\*” “\*” *file1 file2*

The wildcard argument, “\*”, when it appears directly after the *--exclude-object-attribute* option, will match all objects in the files being compared.

The wildcard argument, “\*”, when it appears as the second argument after the *--exclude-object-attribute* option, will match all attributes for the specified objects.

This approach can also support two additional use cases, where the wildcard matches all objects in a specific group:

* Case A5: For all objects in a specific group, exclude a specific attribute

h5diff ***--exclude-object-attribute*** “full\_path\_to\_group/\*” “*attribute\_name*” *file1 file2*

* Case A6: For all objects in a specific group, exclude all attributes

h5diff ***--exclude-object-attribute*** “full\_path\_to\_group/\*” “\*” *file1 file2*

If any of the objects in the specific group are group objects, the specified attribute(s) for those groups and for all objects in the hierarchy below those groups will be excluded from comparison. Attributes associated with the group object named by “full\_path\_to\_group” are not excluded from comparison.

For example,

>> h5diff --exclude-object-attribute “/g1/\*” “attr” File1.h5 File2.h5

would exclude from comparison all attributes named “attr” that are associated with objects that are under the /g1/ group hierarchy in either File1.h5 or File2.h5. If there is an attribute “attr” associated with the /g1 group objects, those would not be excluded from comparison.

The full pattern matching capabilities of the wildcard character ‘\*’ are not supported. For example,

consider the command:

>> h5diff --exclude-object-attribute “/g\*” “attr\*”

In this example, the first argument identifies the specific object whose path is “/g\*”; it is not a pattern match identifying any object whose path specification starts with the characters “/g”. The second argument identifies the attribute whose name is “attr\*”; it is not a pattern match identifying any attribute whose name starts with the characters “attr”.

Restrictions: With the use of the asterisk wildcard in this approach, two restrictions are imposed on the h5diff capabilities. First, h5diff will not be able to individually exclude comparisons of attributes for a specific object (group, dataset, committed datatype) whose full path ends with an asterisk (\*). Second, it will not be able to individually exclude comparisons of attributes whose full name is “\*”.

For example,

>> h5diff --exclude-object-attribute “/g1/\*” “\*” File1.h5 File2.h5

will be interpreted as case A6 (for all objects in group /g1, exclude all attributes), and not as case A1 (for the specific object whose full path is /g1/\*, exclude the specific attribute named \*).

As the example indicates, although the objects and attributes named \* cannot be specified individually, they will be picked up by the wildcard arguments “\*”, along with all other matching objects and attributes.

#### Approach Two

Provide unique options for each case. The current candidates are shown:

* Case A1: For a specific object, exclude a specific attribute

h5diff ***--exclude-object-attribute*** “*full\_path\_to\_object*” “*attribute\_name*” *file1 file2*

* Case A2: For a specific object, exclude all attributes

h5diff ***--exclude-object-all-attributes*** “*full\_path\_to\_object*” *file1 file2*

* Case A3: For all objects, exclude a specific attribute

h5diff ***--exclude-attribute*** “*attribute\_name*” *file1 file2*

* Case A4: For all objects, exclude all attributes

h5diff ***--exclude-all-attributes*** *file1 file2*

This approach does not have any restrictions on handling HDF5 objects and attributes named “\*”. It also does not support Cases A5 and A6, where attributes on all objects in a group can be excluded from comparison, without excluding comparison of attributes on the group itself.

## Specify Exclude Options in a Text File

The ability to specify the options for excluding target(s)—currently object, and soon attribute(s)—from comparison in a text file seems desirable. This is especially true given the use of h5diff in scripted batch processes where similar exclude options are often repeated, and the potential for multiple exclude options and long path names.

Three candidate names for the option used to specify the text file that contains the exclude commands are currently under consideration; we are actively seeking input regarding the preferred name:

h5diff ***--control-file-exclude*** *text\_file**file1 file2*

h5diff ***--command-file-exclude*** *text\_file file1 file2*

h5diff ***--config-file-exclude*** *text\_file**file1 file2*

The text file can have one or more lines, where each line contains an h5diff exclude command specifying target(s) to be excluded from comparison. The h5diff options are the same regardless of whether they appear on the command line or in the text file, making it easy to move between the two methods of specifying exclusions.

The contents of a sample text file named “exclude\_targets” are shown:

--exclude-object “/gg1”

--exclude-object “/gg2”

--exclude-object “/g1/g2/d2”

--exclude-object “/g1/g2/d4”

--exclude-object-attribute “/g1/g2/d8” “speed high”

An example h5diff command using this text file and the second candidate for the option name is:

>> h5diff --command-file-exclude exclude\_targets FileOne.h5 FileTen.h5

# 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.
* Support multiple paths with one invocation of the *--exclude-object* 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 (Section 4.1) command:

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

could be replaced by:

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

And, the Use Case 3 (Section 4.3) command:

>> h5diff --exclude-object “/gg1” --exclude-object “/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 they enumerate (or even know) the objects that are in only one of the two files.

# Implementation Plan

The --exclude-object option, described in Section 3.1, will be implemented first.

The ability to specify exclude options in a text file, described in Section 5.2, will be implemented next.

Support for *Case A1: For a specific object, exclude a specific attribute*, described in Section 5.1, will then be added. Use cases A2-A6 in Excluding Attribute(s) from comparison are considered optional.

# Acknowledgements

This work was prompted by a request from Mike Linda (SSAI - NPP SDS Ozone PEATE), who frequently wants to use batch scripts 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. |
| June 10, 2010: | Version 4: Incorporated feedback from Mike Linda in Motivation, Current Approach, and Acknowledgements sections.  Changed Version numbering to be consistent with other RFCs (no version 1.x, just version x). |
| *July 9, 2010:* | Version 5 incorporated internal discussions within the HDF Group.  Change --exclude to --exclude-object.  Add chapter 5 and Development procedure sections. |
| *August 26, 2010:* | Version 6 includes clarification of wildcard options, and general editorial updates, including renaming of some sections. |

1. Refer to the h5diff reference manual entry for a more extensive discussion of h5diff’s behavior. [↑](#footnote-ref-1)
2. Note that there may be multiple paths to any given object in an HDF5 file. [↑](#footnote-ref-2)