RFC: Reporting of Non-Comparable Datasets by h5diff

Pedro Vicente (pvn@hdfgroup.org)
Ruth Aydt (aydt@hdfgroup.org)

h5diff does not compare some dataset objects, for a variety of reasons. To improve the usability of h5diff as a tool for identifying non-comparable datasets, this RFC proposes two types of changes. First, it proposes updates to the current definition of non-comparable datasets and to the detailed messages generated for non-comparable datasets to make them more consistent. Second, it proposes an option whereby h5diff will only print messages related to dataset objects that could not be compared.

1 Introduction

h5diff does not compare some dataset objects, for a variety of reasons. When this happens, h5diff prints "Some objects are not comparable" at the end of the program execution. In verbose mode, h5diff also prints the reason(s) why it did not perform a comparison for each non-comparable dataset. However, this detailed information is not easy to find among the potentially large amount of output h5diff can generate in verbose mode.

This RFC proposes 1) updates to the current definition of non-comparable datasets and the detailed messages that are output for those datasets to make them more consistent, and 2) a new option that will cause h5diff to only print messages related to dataset objects that could not be compared. We believe these changes will improve the usability of h5diff as a tool for identifying non-comparable dataset objects.

Section 2 discusses the cases where h5diff currently does not compare objects. Section 3 covers the proposed changes in detail, and includes examples of the proposed behavior. The Appendix contains the proposed h5diff usage.

2 Non-Comparable Datasets and Messages: Current Behavior

As background for the proposed changes, this section describes the reasons why h5diff currently does not compare two dataset objects, and shows the detailed information printed by h5diff in verbose mode for each of the reasons.

The syntax of h5diff is:

```
h5diff [OPTIONS] file1 file2 [object1 [object2 ] ]
```

The –v option enables verbose mode.

Note that h5diff accepts zero, one, or two *objects* on the command line. When run with zero or one *objects* specified, the objects being compared in the two files will have the same name. When run with two *objects* specified, the objects being compared in the two files will have different names. For simplicity, the following sections always use the names *object1* and *object2*. Those names will be equivalent when zero or one *objects* are specified on the h5diff command line.

Whenever any of the dataset objects are not comparable, h5diff prints the summary message shown below before exiting. The last line (Use –v…) is not printed when h5diff is run in verbose mode.

```
------------------------------
Some objects are not comparable
------------------------------
Use -v for a list of objects.
```

2.1 Empty datasets

If either *object1* or *object2* is an empty dataset, h5diff does not compare the dataset objects. In verbose mode, the following message is printed:

`<object1> or <object2> are empty datasets`

Developer Note: The routine *H5Dget_storage_size* returns 0 for empty datasets.

2.2 Different datatype classes or H5T_TIME class or H5T_COMPOUND class

If *object1* and *object2* have different datatype classes (e.g., one is of class H5T_INTEGER and the other is of class H5T_FLOAT), h5diff does not compare the dataset objects. In verbose mode, the following message is printed:

`Comparison not possible: <object1> is of class class1 and <object2> is of class class2`

h5diff also does not compare objects if both objects have the H5T_TIME class. In this case, the following message is printed in verbose mode:

`Comparison not possible: <object1> and <object2> are of class H5T_TIME`

h5diff currently does not handle two objects that both have the H5T_COMPOUND datatype class and have non-comparable members.

The possible dataset *classes* are: H5T_INTEGER, H5T_FLOAT, H5T_COMPOUND, H5T_STRING, H5T_ARRAY, H5T_BITFIELD, H5T_OPAQUE, H5T_ENUM, H5T_VLEN, H5T_REFERENCE, and H5T_TIME.

Developer Note: The routine *H5Tget_class* returns the datatype class.

2.3 Different dataspace ranks

If *object1* and *object2* have different dataspace ranks, h5diff does not compare the dataset objects. In verbose mode, the following message is printed:

`Comparison not supported: <object1> has rank rank1, dimensions [dimensions1], max dimensions [dimensions1_max], <object2> has rank rank2, dimensions [dimensions2], max dimensions [dimensions2_max]`

Developer Note: The routine *H5Sget_simple_extent_ndims* returns the dataspace rank.

2.4 Different dataspace dimensions

If *object1* and *object2* have different dataspace current dimensions, h5diff does not compare the dataset objects. In verbose mode, the same message as in case 2.3 is printed. Different maximum dimensions do not invalidate the dataset comparison, but do cause the following message to be printed in verbose mode:

```
Warning: different maximum dimensions
<object1> has max dimensions [dimensions1]
<object2> has max dimensions [dimensions2]
```

Developer Note: The routine *H5Sget_simple_extent_dims* retrieves the current and maximum dataspace dimensions.

2.5 Different sign properties

If *object1* and *object2* have integer datatypes with different sign properties, h5diff does not compare the dataset objects. In verbose mode, the following message is printed:

```
Comparison not supported: <object1> has sign sign1, and <object2> has sign
sign2
```

The possible *signs* are H5T_SGN_NONE and H5T_SGN_2.

Developer Note: The routine *H5Dget_sign* returns the sign property.

2.6 Invalid numeric operation in relative error calculation

The non-comparable datasets behavior described in this section differs from that in the previous sections because it only occurs when h5diff is called with the –p option.

When h5diff is called with the –p option to specify a relative error threshold, it is possible that the relative error calculation could result in a divide by zero. Currently a divide by zero in the relative error calculation causes h5diff to print “Some objects are not comparable” at the end of the program execution. When called with both the –p and the –v options, a message of “not comparable” is printed with the datasets values in the relative column of the h5diff output as shown:

```
$ h5diff –p .10 file1.h5 file2.h5 /g1/dset5 /g1/dset6

dataset: </g1/dset5> and </g1/dset6>
size:           [3x2]           [3x2]
position        dset5           dset6           difference      relative
--------------------------------------------------------------------------
[ 0 0 ]          100             120             20              0.200000
[ 0 1 ]          100             80              20              0.200000
[ 2 1 ]          0               100             100             not comparable
3 differences found
-------------------------------
Some objects are not comparable
-------------------------------
```

Note: The relative error calculation used is |(b-a)/a|, where a and b are numeric values in the first and second datasets.

3 Reporting of Non-Comparable Datasets: Proposed Behavior

This RFC was prompted by a request that h5diff provide complete information about non-comparable datasets independently from the other information it outputs in verbose mode.

In the process of writing the RFC, we noticed some inconsistencies in the definition of non-comparable datasets, and in the detailed messages that are output by h5diff when it detects non-comparable dataset objects. Therefore, the RFC also proposes changes to address those inconsistencies.

In the sections that follow, we 1) propose updates to the current definition of non-comparable datasets and the detailed messages that are output by h5diff, 2) propose a new output format, called *non-comparable list mode*, that only contains information about non-comparable datasets, and 3) show examples of current and proposed h5diff output.

3.1 Proposed behavior and messages for non-comparable datasets

Please refer to the prior sections (2.1 – 2.6) for descriptions of the reasons why two datasets are currently treated as non-comparable. This section details the proposed changes to the h5diff behavior and messages for non-comparable datasets.

Whenever any of the dataset objects are not comparable, h5diff would print the summary message shown below before exiting. The last line (Use –c…) would not be printed when h5diff is run in non-comparable list modes.

```
--------------------------------
Some objects are not comparable
--------------------------------
Use -c for a list of non-comparable objects.
```

The proposed changes to the detailed information about why comparisons were not performed (sections 3.1.1 – 3.1.5) introduce the phrase “Not comparable:” at the beginning of each message. The use of this common string should make it easy for all messages related to non-comparable datasets to be identified using pattern-matching tools such as *grep* and *perl,* even when those messages are embedded in the copious output produced by h5diff in verbose mode. While the proposed non-comparable list mode (section 3.2) should serve many users well, the ability to examine both the verbose and non-comparable dataset information without having to rerun h5diff could save time in some circumstances.

3.1.1 Empty datasets

In verbose mode and non-comparable list mode, the following message would be printed:

```
Not comparable: <object1> or <object2> is an empty dataset
```

3.1.2 Different datatype classes or H5T_TIME class or H5T_COMPOUND class

In verbose mode and non-comparable list mode, the following message would be printed for objects with different datatype classes:

```
Not comparable: <object1> is of class class1 and <object2> is of class class2
```

In verbose mode and non-comparable list mode, the following message would be printed for two objects that both have the H5T_TIME datatype class:

> Not comparable: <*object1*> and <*object2*> are of class H5T_TIME

Under the proposed changes, h5diff would be modified to detect and report when two objects that both have the H5T_COMPOUND datatype class are non-comparable.

If *object1* and *object2* both have the H5T_COMPOUND datatype class, but have different number of fields, h5diff would not compare the dataset objects. In verbose mode and non-comparable list mode, the following message would be printed:

> Not comparable: <*object1*> has *N* fields and <*object2*> has *M* fields

If *object1* and *object2* both have the H5T_COMPOUND datatype class, but fields at the same index have non-comparable datatype classes (e.g., one is of class H5T_INTEGER and the other is of class H5T_FLOAT), h5diff would not compare the dataset objects. In verbose mode and non-comparable list mode, the following message would be printed:

> Not comparable: <*object1*> has a field *field1* with class *class1* and <*object2*> has a field *field2* with class *class2*

Similarly, if fields at the same index of compound types (or nested compound types) have different dataspace ranks, dataspace dimensions, or sign properties, h5diff would not compare *object1* and *object2,* and detailed messages would be printed in verbose and non-comparable list mode. The messages would be analogous to those shown in sections 3.1.3, 3.1.4, and 3.1.5, but instead of "<*objectN*> *has* …" the messages would be phrased "<*objectN*> has a field *fieldM* …"

Developer Note: The routine *H5Tget_nmembers* returns the number of fields in a compound datatype. The routine *H5Tget_member_class* returns the datatype class of a compound datatype field.

3.1.3 Different dataspace ranks

In verbose mode and non-comparable list mode, the following message would be printed:

> Not comparable: <*object1*> has rank *rank1*, dimensions [*dimensions1*], max dimensions [*max_dimensions1*], and <*object2*> has rank *rank2*, dimensions [*dimensions2*], max dimensions [*max_dimensions2*]

3.1.4 Different dataspace dimensions

In verbose mode and non-comparable list mode, the following message would be printed if the current dimensions of the dataspace were different:

> Not comparable: <*object1*> has rank *rank1*, dimensions [*dimensions1*], max dimensions [*max_dimensions1*], and <*object2*> has rank *rank2*, dimensions [*dimensions2*], max dimensions [*max_dimensions2*]

As is currently the case, different maximum dimensions would not invalidate the dataset comparison. The warning message would be printed in verbose mode, but not in non-comparable list mode.

3.1.5 Different sign properties

In verbose mode and non-comparable list mode, the following message would be printed:

```
Not comparable: <object1> has sign sign1 and <object2> has sign sign2
```

3.1.6 Invalid numeric operation in relative error calculation

Invalid numeric operations in relative error calculations would not trigger the "Some objects are not comparable" summary message at the end of the h5diff execution. In addition, the "not comparable" message output by h5diff in the relative column with the –p and –v options would be changed to "divide by zero".

3.2 Proposed output format: non-comparable list mode

We propose the addition of the –c option to h5diff to select *non-comparable list mode.* In this mode, h5diff would only print messages related to dataset objects that could not be compared. The non-comparable list mode would print the detailed information about non-comparable dataset objects, but would not generate the other output produced in verbose mode.

3.3 h5diff output

In this section, two files are compared using h5diff with various options. Current and proposed output is shown.

3.3.1 The files to be compared

The file contents are first revealed using h5dump with the –d option to dump the specified datasets.

```
$ h5dump –d g2/dset1 h5diff_basic2.h5

HDF5 "h5diff_basic2.h5" {
DATASET "g2/dset1" {
   DATATYPE  H5T_IEEE_F64LE
   DATASPACE  SIMPLE { ( 6 ) / ( 6 ) }
   DATA {
   (0): 0, 0, 0, 0, 0, 0
   }
}
}

$ h5dump –d g2/dset2 h5diff_basic2.h5

HDF5 "h5diff_basic2.h5" {
DATASET "g2/dset2" {
   DATATYPE  H5T_IEEE_F64LE
   DATASPACE  SIMPLE { ( 6 ) / ( 6 ) }
   DATA {
   (0): 0, 0, 0, 0, 0, 0
   }
}
}


$ h5dump –d g2/dset3 h5diff_basic2.h5

HDF5 "h5diff_basic2.h5" {
DATASET "g2/dset3" {
   DATATYPE  H5T_STD_I32LE
   DATASPACE  SIMPLE { ( 6 ) / ( 6 ) }
```

```
   DATA {
   (0): 0, 0, 0, 0, 0, 0
   }
}
}

$ h5dump -d g2/dset4 h5diff_basic2.h5

HDF5 "h5diff_basic2.h5" {
DATASET "g2/dset4" {
   DATATYPE  H5T_STD_I32LE
   DATASPACE  SIMPLE { ( 3, 2 ) / ( 3, 2 ) }
   DATA {
   (0,0): 0, 0,
   (1,0): 0, 0,
   (2,0): 0, 0

   }

}

}


$ h5dump -d g2/dset5 h5diff_basic2.h5

HDF5 "h5diff_basic2.h5" {
   DATASET "g2/dset5" {
   DATATYPE  H5T_STD_I32LE
   DATASPACE  SIMPLE { ( 2, 2 ) / ( 2, 2 ) }
   DATA {
   (0,0): 0, 0,
   (1,0): 0, 0

   }

}

}

$ h5dump -d g2/dset6 h5diff_basic2.h5

HDF5 "h5diff_basic2.h5" {
   DATASET "g2/dset6" {
   DATATYPE  H5T_STD_U32LE
   DATASPACE  SIMPLE { ( 3, 2 ) / ( 3, 2 ) }
   DATA {
   (0,0): 0, 0,
   (1,0): 0, 0,
   (2,0): 0, 0

   }

}

}
```

3.3.2 Current output

3.3.2.1 No options

```
$ h5diff h5diff_basic2.h5 h5diff_basic2.h5

--------------------------------
Some objects are not comparable
--------------------------------
Use -v for a list of objects.
```

3.3.2.2 Verbose mode

```
$ h5diff –v h5diff_basic2.h5 h5diff_basic2.h5 obj1 obj2

where obj1 and obj2 are datasets as shown below
```

For the case of empty datasets:

```
dataset: </g2/dset1> and </g2/dset2>
</g2/dset1> or </g2/dset2> are empty datasets
0 differences found
--------------------------------
Some objects are not comparable
--------------------------------
```

For the case of different classes:

```
dataset: </g2/dset2> and </g2/dset3>
Comparison not possible: </g2/dset2> is of class H5T_FLOAT and </g2/dset3> is
of class H5T_INTEGER
0 differences found
--------------------------------
Some objects are not comparable
--------------------------------
```

For the case of different signs:

```
dataset: </g2/dset5> and </g2/dset6>
Comparison not supported: </g2/dset5> has sign H5T_SGN_2 a
nd </g2/dset6> has sign H5T_SGN_NONE
0 differences found
--------------------------------
Some objects are not comparable
--------------------------------
```

For the case of different ranks or dimensions:

```
dataset: </g2/dset3> and </g2/dset4>
Comparison not supported: </g2/dset3> has rank 1, dimensions [6], max
dimensions [6]
</g2/dset4> has rank 2, dimensions [3x2], max dimensions [3x2]0 differences
found
--------------------------------
Some objects are not comparable
--------------------------------
```

3.3.3 Proposed output

3.3.3.1 No options

```
$ h5diff h5diff_basic2.h5 h5diff_basic2.h5

--------------------------------
Some objects are not comparable
--------------------------------
Use -c for a list of objects.
```

3.3.3.2 Verbose mode

```
$ h5diff –v file1 file2 obj1 obj2

where obj1 and obj2 are datasets as shown below
```

For the case of empty datasets:

```
dataset: </g2/dset1> and </g2/dset2>
Not comparable: </g2/dset1> or </g2/dset2> is an empty dataset
0 differences found
--------------------------------
Some objects are not comparable
--------------------------------
Use -c for a list of objects.
```

For the case of different classes:

```
dataset: </g2/dset2> and </g2/dset3>
Not comparable: </g2/dset2> is of class H5T_FLOAT and </g2/dset3> is of class
H5T_INTEGER
0 differences found
--------------------------------
Some objects are not comparable
--------------------------------
Use -c for a list of objects.
```

For the case of different signs:

```
dataset: </g2/dset5> and </g2/dset6>
Not comparable: </g2/dset5> has sign H5T_SGN_2 and </g2/dset6> has sign
H5T_SGN_NONE
0 differences found
--------------------------------
Some objects are not comparable
--------------------------------
Use -c for a list of objects.
```

For the case of different ranks or dimensions:

```
dataset: </g2/dset3> and </g2/dset4>
Not comparable: </g2/dset3> has rank 1, dimensions [6], max dimensions [6]
and </g2/dset4> has rank 2, dimensions [3x2], max dimensions [3x2]
0 differences found
--------------------------------
Some objects are not comparable
```

```
------------------------------
Use -c for a list of objects.
```

3.3.3.3 Non-comparable list mode

```
$ h5diff –c h5diff_basic2.h5 h5diff_basic2.h5 obj1 obj2
```

where *obj1* and *obj2* are datasets as shown below

For the case of empty datasets:

```
Not comparable: </g2/dset1> or </g2/dset2> is an empty dataset
```

For the case of different classes:

```
</g2/dset2> is of class H5T_FLOAT and </g2/dset3> is of class H5T_INTEGER
```

For the case of different signs:

```
Not comparable: </g2/dset5> has sign H5T_SGN_2 and </g2/dset6> has sign
H5T_SGN_NONE
```

For the case of different ranks or dimensions:

```
Not comparable: </g2/dset3> has rank 1, dimensions [6], max dimensions [6]
and </g2/dset4> has rank 2, dimensions [3x2], max dimensions [3x2]
```

Acknowledgements

This work was inspired by comments from Cheryl Craig (cacraig@ucar.edu), National Center for Atmospheric Research, Atmospheric Chemistry Division.

Revision History

December 18, 2008:	Version 1 circulated for comment within The HDF Group.
December 30, 2008:	Version 2 incorporates suggestions from Mike Folk. (mfolk@hdfgroup.org)
February 09, 2009:	Version 3 incorporates suggestions from Ruth Aydt. (aydt@hdfgroup.org)
February 19, 2009:	Version 4 incorporates suggestions from Ruth Aydt.
April 13, 2009:	Version 5 circulated for comments within The HDF Group
May 27, 2009:	Version 6 includes updates to Section 3.3 and its subsections. Published for public comment. Comments should be sent to pvn@hdfgroup.org or aydt@hdfgroup.org or help@hdfgroup.org .

Appendix: Proposed h5diff Usage

The proposed h5diff usage information is shown:

```
Usage: h5diff [OPTIONS] file1 file2 [obj1[obj2]]
Compare objects in HDF5 files

  file1                    Name of the first HDF5 file
  file2                    Name of the second HDF5 file
  [obj1]                   Name of an HDF5 object in file1, using absolute path
  [obj2]                   Name of an HDF5 object in file2, using absolute path
  OPTIONS
   -h, --help              Print usage message and exit
   -V, --version           Print version number and exit
   -r, --report            Report mode. Print differences
   -v, --verbose           Verbose mode. Print differences, list of objects
   -q, --quiet             Quiet mode. No output printed
   -c, --compare           Output details for objects that are not comparable
   -n C, --count=C         Print first C differences
   -d D, --delta=D         Print numeric difference when greater than limit D
   -p R, --relative=R      Print numeric difference when greater than relative limit R

  C - is a positive integer
  D - is a positive number. Compare criteria is |a - b| > D
  R - is a positive number. Compare criteria is |(b-a)/a| > R

 Modes of output:

  Default mode: Print the number of differences found and where they occurred
  -r Report mode: Default mode output plus the differences
  -v Verbose mode: Report mode output plus a list of objects and warnings
  -c Non-comparable list mode: Print reasons why objects can't be compared
  -q Quiet mode: No output printed

 Comparison criteria:

 If no objects [obj1[obj2]] are specified, h5diff only compares objects
   with the same absolute path in both files.

 The comparison criteria varies for different types of objects:
   1) datasets: compare values of of array elements
   2) groups: string comparison of group names
   3) datatypes: compare the values returned by H5Tequal
   4) soft links: string comparison of soft link names

 Return values:

  0 if no differences; 1 if differences found; 2 if error

 Usage examples:

 1) h5diff file1 file2 /g1/dset1 /g1/dset2
```

```
   Compares object '/g1/dset1' in file1 with '/g1/dset2' in file2

2) h5diff file1 file2 /g1/dset1

   Compares object '/g1/dset1' in both files

3) h5diff file1 file2

   Compares all objects in both files

Note:  file1 and file2 can be the same file. Use

   h5diff file1 file1 /g1/dset1 /g1/dset2

   to compare '/g1/dset1' and '/g1/dset2' in the same file
```