# The h5diff’s current behaviors and shortcomings

## Jonathan Kim

## Purpose

To list h5diff’s current behaviors, so sensible and nonsensical behaviors can be figured out. With these results, we can prevent those issues from the new h5compare tool and decide what to do with current h5diff.

## Background

Many features have been implemented into the current h5diff command tool over its lifetime. However when some of these changes were made, they failed to address related updates that must go along with the feature (ex: default output or exit code and so on). Also, a lack of fundamental definitions contributed to other inconsistent behaviors in a narrow scope of development procedures. As a result, the current h5diff tool implements some inconsistent behaviors which confuse users. This document can also be used during development of a new HDF5 file comparison tool, as a reference for comparing behaviors of the old and new tools in sensible manner.

## Overview

There are four main sections to show the current h5diff tool’s behaviors. A section may contain overlapped contents from other section.

1. Default behaviors - h5diff’s core behaviors
2. Optional behaviors - additional or exceptional behaviors
3. How to handle common, extra or not-comparable object and attribute with results - what user would look and feel in a big picture
4. Other behaviors - other known issues (this can grow as we identify more issues)

All of these will contain example output and exit-code from h5diff, so it can be verified together.

**Highlights indicate either insensible or inconsistent results at current stage.**

* Color – insensible behavior.
* Color – need to think about.

## Example HDF5 files

All the HDF5 files that are used in each section can be found in “Appendix A”, so refer to specific files to verify the h5diff’s behaviors from examples in each section.

Table of Contents

[1. Default behaviors 4](#_Toc309308307)

[1.1 Dataset : *Number* 4](#_Toc309308308)

[1.2 Dataset : *String* 4](#_Toc309308309)

[1.3 Dataset: Space array 5](#_Toc309308310)

[1.4 Dataset: Type array 6](#_Toc309308311)

[1.5 Dataset: Object reference 7](#_Toc309308312)

[1.6 Dataset: Region reference 9](#_Toc309308313)

[1.7 Dataset: *Empty* 10](#_Toc309308314)

[1.8 Dataset: Invalid enum value 10](#_Toc309308315)

[1.9 Datatype: Different type 11](#_Toc309308316)

[1.10 Group: Recursive by default 12](#_Toc309308317)

[2. Optional behaviors 13](#_Toc309308318)

[2.1 ‘-n C’ or ‘--count=C’ 13](#_Toc309308319)

[2.2 ‘-d D’ or ‘--delta=D’ 14](#_Toc309308320)

[2.3 ‘-p R’ or ‘–relative=R’ 15](#_Toc309308321)

[2.4 --use-system-epsilon 16](#_Toc309308322)

[2.5 --exclude-path “path” 17](#_Toc309308323)

[2.6 --follow-symlinks (softlink as an example) 18](#_Toc309308324)

[2.7 --no-dangling-links 20](#_Toc309308325)

[2.8 ‘-N’ or ‘--nan’ 22](#_Toc309308326)

[2.9 ‘-v1’ and ‘-v2’ 22](#_Toc309308327)

[2.10 ‘-c’ or ‘--compare’ 22](#_Toc309308328)

[3. Handle common, extra or non-comparable object and attribute with results 23](#_Toc309308329)

[3.1 When difference found in Common object or attribute 24](#_Toc309308332)

[3.2 When a difference is found because an Extra object or attribute exists 24](#_Toc309308333)

[3.2.1 When extra object exists in a file 24](#_Toc309308334)

[3.2.2 When extra attribute exists 25](#_Toc309308335)

[3.3 When difference found in Non-comparable object or attribute 27](#_Toc309308336)

[3.3.1 Example for non-comparable object 28](#_Toc309308337)

[3.3.2 Example for mixed type for common object (missing code) 29](#_Toc309308338)

[4. Other behaviors 30](#_Toc309308339)

[4.1 Empty file vs. Non empty file 30](#_Toc309308341)

[Appendix A – List of example HDF5 files (in alphabetic order) 31](#_Toc309308342)

[Acknowledgements 43](#_Toc309308343)

[Revision History 43](#_Toc309308344)

## Default behaviors

### Dataset : *Number*

|  |  |
| --- | --- |
| **Compare *Object* vs *Object***  (ex: $h5diff file1.h5 file2.h5 /dset1 /dset2) | **Compare *File* vs *File***  (ex: $h5diff file1.h5 file2.h5) |
| $ **h5diff dset-num1\_1.h5 dset-num1\_2.h5 /dset**  dataset: </dset> and </dset>  4 differences found | $ **h5diff dset-num1\_1.h5 dset-num1\_2.h5**  dataset: </dset> and </dset>  4 differences found |
| $ **h5diff -v dset-num1\_1.h5 dset-num1\_2.h5 /dset**  dataset: </dset> and </dset>  size: [2x2] [2x2]  position dset dset difference  ------------------------------------------------------------  [ 0 0 ] 1 0 1  [ 0 1 ] 2 0 2  [ 1 0 ] 3 0 3  [ 1 1 ] 4 0 4  4 differences found | $ **h5diff -v dset-num1\_1.h5 dset-num1\_2.h5**  file1 file2  ---------------------------------------  x x /  x x /dset  group : </> and </>  0 differences found  dataset: </dset> and </dset>  size: [2x2] [2x2]  position dset dset difference  ------------------------------------------------------------  [ 0 0 ] 1 0 1  [ 0 1 ] 2 0 2  [ 1 0 ] 3 0 3  [ 1 1 ] 4 0 4  4 differences found |
| EXIT-CODE : 1 | EXIT-CODE : 1 |

### Dataset : *String*

|  |  |
| --- | --- |
| **Compare *Object* vs *Object*** | **Compare *File* vs *File*** |
| $ **h5diff dset-str1\_1.h5 dset-str1\_2.h5 /string**  dataset: </string> and </string>  6 differences found | $ **h5diff dset-str1\_1.h5 dset-str1\_2.h5**  dataset: </string> and </string>  6 differences found |
| $ **h5diff -v dset-str1\_1.h5 dset-str1\_2.h5 /string**  dataset: </string> and </string>  size: [2] [2]  position string string difference  ------------------------------------------------------------  [ 0 ] y c  [ 0 ] y a  [ 0 ] y t  [ 1 ] z d  [ 1 ] z o  [ 1 ] z g  6 differences found | $ **h5diff -v dset-str1\_1.h5 dset-str1\_2.h5**  file1 file2  ---------------------------------------  x x /  x x /string  group : </> and </>  0 differences found  dataset: </string> and </string>  size: [2] [2]  position string string difference  ------------------------------------------------------------  [ 0 ] y c  [ 0 ] y a  [ 0 ] y t  [ 1 ] z d  [ 1 ] z o  [ 1 ] z g  6 differences found |
| EXIT-CODE : 1 | EXIT-CODE : 1 |
| NOTE: currently only compare by single character as a unit. Need to also consider comparing by string as a unit. (user requested) If compared by string as a unit, there would be 2 differences. One from “yyy/cat” the other from “zzz/dog” | |
| DISCUSSION: | |

### Dataset: Space array

|  |  |
| --- | --- |
| **Compare *Object* vs *Object*** | **Compare *File* vs *File*** |
| **$ h5diff dset-space-array1.h5 dset-space-array2.h5 /space-array**  dataset: </space-array> and </space-array>  3 differences found | **$ h5diff dset-space-array1.h5 dset-space-array2.h5**  dataset: </space-array> and </space-array>  3 differences found |
| **$ h5diff -v dset-space-array1.h5 dset-space-array2.h5 /space-array**  dataset: </space-array> and </space-array>  size: [2x3] [2x3]  position space-array space-array difference  ---------------------------------------------------------  [ 0 1 ] 0 10 10  [ 1 0 ] 1 10 9  [ 1 2 ] 1 10 9  3 differences found | **$ h5diff -v dset-space-array1.h5 dset-space-array2.h5**  file1 file2  ---------------------------------------  x x /  x x /space-array  group : </> and </>  0 differences found  dataset: </space-array> and </space-array>  size: [2x3] [2x3]  position space-array space-array difference  ----------------------------------------------------------  [ 0 1 ] 0 10 10  [ 1 0 ] 1 10 9  [ 1 2 ] 1 10 9  3 differences found |
| EXIT-CODE : 1 | EXIT-CODE : 1 |

### Dataset: Type array

|  |  |
| --- | --- |
| **Compare *Object* vs *Object*** | **Compare *File* vs *File*** |
| **$ h5diff dset-type-array1.h5 dset-type-array2.h5 /type-array**  dataset: </type-array> and </type-array>  3 differences found | **$ h5diff dset-type-array1.h5 dset-type-array2.h5**  dataset: </type-array> and </type-array>  3 differences found |
| **$ h5diff -v dset-type-array1.h5 dset-type-array2.h5 /type-array**  dataset: </type-array> and </type-array>  size: [2] [2]  position type-array type-array difference  ----------------------------------------------------------  [ 0 ] 0 10 10  [ 1 ] 1 10 9  [ 1 ] 1 10 9  3 differences found | **$ h5diff -v dset-type-array1.h5 dset-type-array2.h5**  file1 file2  ---------------------------------------  x x /  x x /type-array  group : </> and </>  0 differences found  dataset: </type-array> and </type-array>  size: [2] [2]  position type-array type-array difference  ----------------------------------------------------------  [ 0 ] 0 10 10  [ 1 ] 1 10 9  [ 1 ] 1 10 9  3 differences found |
| EXIT-CODE : 1 | EXIT-CODE : 1 |
| NOTE: currently only compare by single value as a unit. Need to also consider comparing by set of values as a unit. (same concept as char vs string) If compare by set of values in this example, there would be 2 differences. One from (0): , other from (1): . | |
| DISCUSSION: | |

### Dataset: Object reference

|  |  |
| --- | --- |
| **Compare *Object* vs *Object*** | **Compare *File* vs *File*** |
| **$ h5diff ref-obj1.h5 ref-obj2.h5 /Dset\_OBJREF**  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects. | **$ h5diff ref-obj1.h5 ref-obj2.h5**  dataset: </Dset1> and </Dset1>  1 differences found  datatype: </NamedDatatype> and </NamedDatatype>  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects. |
| **$ h5diff -c ref-obj1.h5 ref-obj2.h5 /Dset\_OBJREF**  <NONE> | **$ h5diff -c ref-obj1.h5 ref-obj2.h5**  dataset: </Dset1> and </Dset1>  1 differences found  datatype: </NamedDatatype> and </NamedDatatype>  <no display for why not comparable> |
| ISSUE: -c didn’t display any other message. | |
|  | ISSUE: Didn’t display difference for /Group/Dset2. However the difference is displayed with -v.  ISSUE: Didn’t display ”1 differences found” for NamedDatatype. ( no code exist) However the difference is displayed with –v. |
| EXIT-CODE : 1 | EXIT-CODE : 1 |
| **$ h5diff -v ref-obj1.h5 ref-obj2.h5 /Dset\_OBJREF**  dataset: </Dset\_OBJREF> and </Dset\_OBJREF>  size: [3] [3]  position difference  ----------------------------------------------------------  [ 0 ] 0 1 1  Warning: Comparison not possible of object types referenced: <Dset\_OBJREF> and <Dset\_OBJREF>  Warning: Comparison not possible of object types referenced: <Dset\_OBJREF> and <Dset\_OBJREF>  1 differences found  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects. | **$ h5diff -v ref-obj1.h5 ref-obj2.h5**  file1 file2  ---------------------------------------  x x /  x x /Dset1  x x /Dset\_OBJREF  x x /Group  x x /Group/Dset2  x x /NamedDatatype  group : </> and </>  0 differences found  dataset: </Dset1> and </Dset1>  size: [3] [3]  position Dset1 Dset1 difference  ----------------------------------------------------------  [ 0 ] 0 1 1  1 differences found  dataset: </Dset\_OBJREF> and </Dset\_OBJREF>  size: [3] [3]  position difference  ----------------------------------------------------------  [ 0 ] 0 1 1  Warning: Comparison not possible of object types referenced: <Dset\_OBJREF> and <Dset\_OBJREF>  Warning: Comparison not possible of object types referenced: <Dset\_OBJREF> and <Dset\_OBJREF>  1 differences found  group : </Group> and </Group>  0 differences found  dataset: </Group/Dset2> and </Group/Dset2>  size: [3] [3]  position Dset2 Dset2 difference  ----------------------------------------------------------  [ 2 ] 0 2 2  1 differences found  datatype: </NamedDatatype> and </NamedDatatype>  1 differences found  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects. |
| EXIT-CODE : 1 | EXIT-CODE : 1 |
| ISSUE: It seems that the 1st warning is from obj-ref to “/Group” and the 2nd warning is from obj-ref to “/NamedDatatype”, there are expected to be comparable because the both obj references points same object names. The first file was copied and some values were changed, so should display the differences from the object references. | |

### Dataset: Region reference

|  |  |
| --- | --- |
| **Compare *Object* vs *Object*** | **Compare *File* vs *File*** |
| **$ h5diff ref-dsetreg1.h5 ref-dsetreg2.h5 /Dset\_REGREF**  <NONE> | **$ h5diff ref-dsetreg1.h5 ref-dsetreg2.h5**  dataset: </dset> and </dset>  48 differences found |
| **$ h5diff -v ref-dsetreg1.h5 ref-dsetreg2.h5 /Dset\_REGREF**  dataset: </Dset\_REGREF> and </Dset\_REGREF>  0 differences found | **$ h5diff -v ref-dsetreg1.h5 ref-dsetreg2.h5**  file1 file2  ---------------------------------------  x x /  x x /Dset\_REGREF  x x /dset  group : </> and </>  0 differences found  dataset: </Dset\_REGREF> and </Dset\_REGREF>  0 differences found  dataset: </dset> and </dset>  size: [3x16] [3x16]  position dset dset difference  ----------------------------------------------------------  [ 0 0 ] 0 1 1  . . .  [ 2 15 ] 0 3 3  48 differences found |
| EXIT-CODE : 0 | EXIT-CODE : 1 (but not from region reference dataset) |
| ISSUE: Didn’t follow to compare region reference’s end point values. | |

### Dataset: *Empty*

|  |  |
| --- | --- |
| **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| **$ h5diff dset\_empty1.h5 dset\_empty2.h5 /empty\_d1**  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects. | **$ h5diff dset\_empty1.h5 dset\_empty2.h5**  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects. |
| **$ h5diff -c dset\_empty1.h5 dset\_empty2.h5 /empty\_d1**  Not comparable: </empty\_d1> or </empty\_d1> is an empty dataset | **$ h5diff -c dset\_empty1.h5 dset\_empty2.h5**  Not comparable: </empty\_d1> or </empty\_d1> is an empty dataset  Not comparable: </empty\_d2> or </empty\_d2> is an empty dataset |
| EXIT-CODE : 0 | EXIT-CODE : 0 |
| ISSUE: This empty dataset has same type and space. If type and space is same, empty should not be reason to treat it as non-comparable. It should be viewed as same dataset with empty condition. | |

### Dataset: Invalid enum value

|  |  |
| --- | --- |
| **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| **$ h5diff enum\_invalid.h5 enum\_invalid.h5 /dset1 /dset2**  dataset: </dset1> and </dset2>  3 differences found | NO NEED |
| **$ h5diff -v enum\_invalid.h5 enum\_invalid.h5 /dset1 /dset2**  dataset: </dset1> and </dset2>  size: [6] [6]  position dset1 dset2 difference  ------------------------------------------------------------  [ 1 ] YIN \*\*INVALID VALUE\*\*  [ 2 ] \*\*INVALID VALUE\*\* YIN  [ 5 ] YIN YANG  3 differences found |  |
| EXIT-CODE : 1 |  |
| ISSUE: There are two pair of invalid enum values. We may consider to display 5 differences. So user can aware the status of invalid values.  Refer to “enum\_invalid.h5” in APPENDIX A | |

### Datatype: Different type

|  |  |
| --- | --- |
| **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| **$ h5diff ref-obj1.h5 ref-obj2.h5 /NamedDatatype /NamedDatatype**  datatype: </NamedDatatype> and </NamedDatatype>  <no output of difference found> | NO NEED |
| **$ h5diff -v ref-obj1.h5 ref-obj2.h5 /NamedDatatype /NamedDatatype**  datatype: </NamedDatatype> and </NamedDatatype>  1 differences found |
| EXIT-CODE : 1 |  |
| ISSUE: Expect to see “1 differences found” also for default. | |

### Group: Recursive by default

|  |  |
| --- | --- |
| **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| **$ h5diff -v groups1.h5 groups2.h5 /grp1 /grp1**  group1 group2  ---------------------------------------  x x  x x /dset  x x /grp2  x x /grp2/dset  group : </grp1> and </grp1>  0 differences found  dataset: </grp1/dset> and </grp1/dset>  size: [3] [3]  position dset dset difference  ------------------------------------------------------------  [ 1 ] 0 1 1  1 differences found  group : </grp1/grp2> and </grp1/grp2>  0 differences found  dataset: </grp1/grp2/dset> and </grp1/grp2/dset>  size: [3] [3]  position dset dset difference  ------------------------------------------------------------  [ 1 ] 0 2 2  1 differences found | **$ h5diff -v groups1.h5 groups2.h5**  file1 file2  ---------------------------------------  x x /  x x /grp1  x x /grp1/dset  x x /grp1/grp2  x x /grp1/grp2/dset  group : </> and </>  0 differences found  group : </grp1> and </grp1>  0 differences found  dataset: </grp1/dset> and </grp1/dset>  size: [3] [3]  position dset dset difference  ------------------------------------------------------------  [ 1 ] 0 1 1  1 differences found  group : </grp1/grp2> and </grp1/grp2>  0 differences found  dataset: </grp1/grp2/dset> and </grp1/grp2/dset>  size: [3] [3]  position dset dset difference  ------------------------------------------------------------  [ 1 ] 0 2 2  1 differences found |
| EXIT-CODE : 1 | EXIT-CODE : 1 |
| NOTE: Blank line between object would be easier to read the output. (should consider for h5compare’s new level output) | |

## Optional behaviors

### ‘-n C’ or ‘--count=C’

: Print differences up to C number, C is a positive integer.

|  |  |
| --- | --- |
| **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| Not necessary. | **$ h5diff -v dset-num1\_1.h5 dset-num1\_2.h5**  file1 file2  ---------------------------------------  x x /  x x /dset  group : </> and </>  0 differences found  dataset: </dset> and </dset>  size: [2x2] [2x2]  position dset dset difference  ------------------------------------------------------------  [ 0 0 ] 1 0 1  [ 0 1 ] 2 0 2  [ 1 0 ] 3 0 3  [ 1 1 ] 4 0 4  4 differences found |
| **$ h5diff -v -n 2 dset-num1\_1.h5 dset-num1\_2.h5**  file1 file2  ---------------------------------------  x x /  x x /dset  group : </> and </>  0 differences found  dataset: </dset> and </dset>  size: [2x2] [2x2]  position dset dset difference  ------------------------------------------------------------  [ 0 0 ] 1 0 1  [ 0 1 ] 2 0 2  2 differences found |
|  | EXIT CODE: 1 |

### ‘-d D’ or ‘--delta=D’

: Print difference if (|a-b| > D). D must be a positive number. Cannot use with '-p' or '--use-system-epsilon'.

|  |  |
| --- | --- |
| **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| **$ h5diff -v dset-num2\_1.h5 dset-num2\_2.h5 /dset1**  dataset: </dset1> and </dset1>  size: [4] [4]  position dset1 dset1 difference  ------------------------------------------------------------  [ 0 ] 0 10 10  [ 1 ] 0 20 20  [ 2 ] 0 30 30  [ 3 ] 0 40 40  4 differences found | Not need (same concept) |
| **$ h5diff -v -d 20 dset-num2\_1.h5 dset-num2\_2.h5 /dset1**  dataset: </dset1> and </dset1>  size: [4] [4]  position dset1 dset1 difference  ------------------------------------------------------------  [ 2 ] 0 30 30  [ 3 ] 0 40 40  2 differences found |
| EXIT CODE: 1 |  |

### ‘-p R’ or ‘–relative=R’

: Print difference if (|(a-b)/b| > R). R must be a positive number. Cannot use with '-d' or '--use-system-epsilon'.

|  |  |
| --- | --- |
| **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| **$ h5diff -v dset-num3\_1.h5 dset-num3\_2.h5 /dset**  dataset: </dset> and </dset>  size: [3x2] [3x2]  position dset dset difference  ------------------------------------------------------------  [ 0 0 ] 100 120 20  [ 0 1 ] 100 80 20  [ 1 0 ] 100 0 100  [ 1 1 ] 0 100 100  [ 2 1 ] 100 50 50  5 differences found  **$ h5diff -v -p 0.05 dset-num3\_1.h5 dset-num3\_2.h5 /dset**  dataset: </dset> and </dset>  size: [3x2] [3x2]  position dset dset difference relative  ------------------------------------------------------------------------  [ 0 0 ] 100 120 20 0.200000  [ 0 1 ] 100 80 20 0.200000  [ 1 0 ] 100 0 100 1.000000  [ 1 1 ] 0 100 100 not comparable  [ 2 1 ] 100 50 50 0.500000  5 differences found  **$ h5diff -v -p 0.3 dset-num3\_1.h5 dset-num3\_2.h5 /dset**  dataset: </dset> and </dset>  size: [3x2] [3x2]  position dset dset difference relative  ------------------------------------------------------------------------  [ 1 0 ] 100 0 100 1.000000  [ 1 1 ] 0 100 100 not comparable  [ 2 1 ] 100 50 50 0.500000  3 differences found | Not need |
| EXIT CODE: 1 |  |

### --use-system-epsilon

TODO

### --exclude-path “path”

: Exclude the specified path to an object when comparing files or groups. Refer to help page or RM for more details.

TODO

### --follow-symlinks (softlink as an example)

|  |  |
| --- | --- |
| : Follow symbolic links (soft links and external links) and compare the links' target objects. Refer to help page or RM for more details.**Compare object vs object** | **Compare *file* vs *file*** |
| Soft link to a dataset | **$ h5diff -v softlinks1.h5 softlinks2.h5**  file1 file2  ---------------------------------------  x x /  x x /dset  x x /grp  x x /grp/gdset  x x /softlink2dset  x x /softlink2grp  group : </> and </>  0 differences found  dataset: </dset> and </dset>  size: [3] [3]  position dset dset difference  ------------------------------------------------------------  [ 0 ] 1 2 1  1 differences found  group : </grp> and </grp>  0 differences found  dataset: </grp/gdset> and </grp/gdset>  size: [3] [3]  position gdset gdset difference  ------------------------------------------------------------  [ 1 ] 1 2 1  1 differences found  link : </softlink2dset> and </softlink2dset>  0 differences found  link : </softlink2grp> and </softlink2grp>  0 differences found  EXIT CODE: 1 |
| **$ h5diff -v softlinks1.h5 softlinks2.h5 /softlink2dset**  link : </softlink2dset> and </softlink2dset>  0 differences found  EXIT CODE: 0 |
| **$ h5diff -v --follow-symlinks softlinks1.h5 softlinks2.h5 /softlink2dset**  dataset: </softlink2dset> and </softlink2dset>  size: [3] [3]  position softlink2dset softlink2dset difference  ------------------------------------------------------------  [ 0 ] 1 2 1  1 differences found  EXIT CODE: 1 |
| Soft link to a group |
| **$ h5diff -v softlinks1.h5 softlinks2.h5 /softlink2grp**  link : </softlink2grp> and </softlink2grp>  0 differences found  EXIT CODE: 0 |
| **$ h5diff -v --follow-symlinks softlinks1.h5 softlinks2.h5 /softlink2grp**  group1 group2  ---------------------------------------  x x  x x /gdset  group : </softlink2grp> and </softlink2grp>  0 differences found  dataset: </softlink2grp/gdset> and </softlink2grp/gdset>  size: [3] [3]  position gdset gdset difference  ---------------------------------------------------------  [ 1 ] 1 2 1  1 differences found  EXIT CODE: 1 |
| **$ h5diff -v --follow-symlinks softlinks1.h5 softlinks2.h5**  file1 file2  ---------------------------------------  x x /  x x /dset  x x /grp  x x /grp/gdset  x x /softlink2dset  x x /softlink2grp  x x /softlink2grp/gdset  group : </> and </>  0 differences found  dataset: </dset> and </dset>  size: [3] [3]  position dset dset difference  ------------------------------------------------------------  [ 0 ] 1 2 1  1 differences found  group : </grp> and </grp>  0 differences found  dataset: </grp/gdset> and </grp/gdset>  size: [3] [3]  position gdset gdset difference  ------------------------------------------------------------  [ 1 ] 1 2 1  1 differences found  dataset: </softlink2dset> and </softlink2dset>  size: [3] [3]  position softlink2dset softlink2dset difference  ------------------------------------------------------------  [ 0 ] 1 2 1  1 differences found  group : </softlink2grp> and </softlink2grp>  0 differences found  dataset: </softlink2grp/gdset> and </softlink2grp/gdset>  size: [3] [3]  position gdset gdset difference  ------------------------------------------------------------  [ 1 ] 1 2 1  1 differences found  EXIT CODE: 1 |

### --no-dangling-links

: must used with ‘--follow-symlinks’ , otherwise display error.

|  |  |
| --- | --- |
| **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| Both dangling links **$ h5diff -v danglelinks1.h5 danglelinks2.h5 /slink1**  dangling link: </slink1> and </slink1>  0 differences found  **$ h5diff -v danglelinks1.h5 danglelinks2.h5 /extlink1**  dangling link: </extlink1> and </extlink1>  0 differences found  **$ h5diff -v danglelinks1.h5 danglelinks2.h5 /extlink2**  dangling link: </extlink2> and </extlink2>  0 differences found  All EXIT CODE: 0 Only one side dangling link **$ h5diff -v danglelinks1.h5 danglelinks2.h5 /slink1 /slink2**  obj1 </slink1> is a dangling link.  1 differences found  **$ h5diff -v danglelinks1.h5 danglelinks2.h5 /extlink3 /extlink1**  obj2 </extlink1> is a dangling link.  1 differences found  All EXIT CODE: 1 | **$ h5diff -v danglelinks1.h5 danglelinks2.h5**  file1 file2  ---------------------------------------  x x /  x /dset  x x /extlink1  x x /extlink2  x x /extlink3  x x /slink1  x x /slink2  group : </> and </>  0 differences found  dangling link: </extlink1> and </extlink1>  0 differences found  dangling link: </extlink2> and </extlink2>  0 differences found  obj2 </extlink3> is a dangling link.  1 differences found  dangling link: </slink1> and </slink1>  0 differences found  obj1 </slink2> is a dangling link.  1 differences found  EXIT CODE: 1 |
| With ‘--no-dangling-links’ **$ h5diff -v --follow-symlinks --no-dangling-links danglelinks1.h5** **danglelinks2.h5 /slink1**  Warning: </slink1> is a dangling link.  *The above rest cases are same. (display the first detected dangling link name)*  All EXIT CODE: 2 Only with ‘—follow-symlinks’ **$ h5diff -v --follow-symlinks danglelinks1.h5 danglelinks2.h5 /slink1**  obj1 </slink1> is a dangling link.  1 differences found  *The above rest cases are same. (display the first detected dangling link name)*  All EXIT CODE: 1 | With ‘--no-dangling-links’ **$ h5diff -v --follow-symlinks --no-dangling-links danglelinks1.h5 danglelinks2.h5**  file1 file2  ---------------------------------------  x x /  x /dset  x x /extlink1  x x /extlink2  x x /extlink3  x x /slink1  x x /slink2  group : </> and </>  0 differences found  Warning: </extlink1> is a dangling link.  Warning: </extlink2> is a dangling link.  Warning: </extlink3> is a dangling link.  Warning: </slink1> is a dangling link.  Warning: </slink2> is a dangling link.  EXIT CODE: 2 |
|  |  |

### ‘-N’ or ‘--nan’

TODO

### ‘-v1’ and ‘-v2’

These options were added to display more details for the attribute status. Refer to ‘extra attribute’ examples in section 3.

### ‘-c’ or ‘--compare’

: List objects that are not comparable

Refer to ‘non-comparable’ examples in section3.

## Handle common, extra or non-comparable object and attribute with results

### Overview:

In this section, there are three categories how user would look & feel with current h5diff in a big picture when expecting differences.

1. When a difference is found in Common object or attribute
2. When a difference is found because an Extra object or attribute exists
3. When a difference is found because a Non-comparable object or attribute exists

Each section shows how h5diff behaves with default and verbose output along with its exit-code.

Definition of terms:

* **Common object**: The absolute path to the object in two files is the same when comparing the two HDF5 files, or the object name within two specified groups is the same, when comparing the two HDF5 objects.
* **Common attribute**: The attribute name is the same for two objects.
* **Extra object**: An object that doesn’t exist in one file when comparing the two HDF5 files; or an object name that doesn’t exist within one of the specified groups, when comparing the two HDF5 objects.
* **Extra attribute**: An attribute name that doesn’t exist on the other object being compared.
* **Non-comparable object**: Two HDF5 objects that have a different datatype or dataspace.
* **Non-comparable attribute**: Two HDF5 attributes that have a different datatype or dataspace.

### When difference found in Common object or attribute

|  |  |  |
| --- | --- | --- |
|  | **Compare object vs object** | **Compare file vs file** |
| **Common**  object or attribute | Display output   * Default (without ‘-v’ option)   + display number of differences with names of object or attribute * ‘–v’ option   + display details of the differences (data values)   + display number of differences with name of object or attribute | Display output   * Default (without ‘–v’ option)   + display number of differences with names of object or attribute * ‘–v’ option   + object status list   + display details of the differences (data values)   + display number of differences with name of object or attribute |
| EXIT-CODE: 1 | EXIT-CODE: 1 |

### When a difference is found because an Extra object or attribute exists

### 3.2.1 When extra object exists in a file

|  |  |  |
| --- | --- | --- |
|  | **Compare object vs object** | **Compare file vs file** |
| **Extra** object | N/A because user specifies obj1 and obj2 directly; no extra object. | Display output   * Default (without verbose option)   + None * Use ‘-v’ to verify   + List of object status on top |
| EXIT-CODE: 1 |
| ISSUE: No output in default, which causes confusion to user. Some message should be displayed indicating there are differences in default. |
|  | | |

|  |  |  |
| --- | --- | --- |
|  | **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| -v  Extra object:  *display details for extra object* | N/A | **$ h5diff extra-obj1.h5 extra-obj2.h5**  <no output> |
| **$ h5diff -v extra-obj1.h5 extra-obj2.h5**  file1 file2  ---------------------------------------  x x /  x x /do  x /mi  x /re  group : </> and </>  0 differences found  dataset: </do> and </do>  0 differences found |
| EXIT-CODE : 1 | |

### 3.2.2 When extra attribute exists

|  |  |  |
| --- | --- | --- |
|  | **Compare object vs object** | **Compare file vs file** |
| **Extra** attribute | Display output   * Default (without verbose option)   + None * Use ‘-v1’ or ‘-v2’ to verify   + attribute status line and list | Display output   * Default (without verbose option)   + None * Use ‘-v1’ or ‘-v2’ to verify   + attribute status line and list |
| EXIT-CODE: 0 ([HDFFV-7643](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7643)) | EXIT-CODE: 0 ([HDFFV-7643](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7643)) |
| ISSUE: No message output indicating the extra attribute(s) as default  ISSUE: The exit code is expected be 1, which indicates the two objects are different. | |

|  |  |  |
| --- | --- | --- |
|  | **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| -v1 / -v2  Attribute:  *display details for extra attribute* | **$ h5diff extra-attr1.h5 extra-attr2.h5 /do**  <no output> | Same for each object |
| **$ h5diff -v extra-attr1.h5 extra-attr2.h5 /do**  dataset: </do> and </do>  attribute: <attr1 of </do>> and <attr1 of </do>>  0 differences found  0 differences found |
| **$ h5diff -v1 extra-attr1.h5 extra-attr2.h5 /do**  dataset: </do> and </do>  **Attributes status: 1 common, 0 only in obj1, 1 only in obj2**  attribute: <attr1 of </do>> and <attr1 of </do>>  0 differences found  0 differences found |
| **$ h5diff -v2 extra-attr1.h5 extra-attr2.h5 /do**  dataset: </do> and </do>  **obj1 obj2**  **--------------------------------------**  **x x attr1**  **x attr2**  **Attributes status: 1 common, 0 only in obj1, 1 only in obj2**  attribute: <attr1 of </do>> and <attr1 of </do>>  0 differences found  0 differences found |
| EXIT-CODE : 0 | |

### When difference found in Non-comparable object or attribute

|  |  |  |
| --- | --- | --- |
|  | **Compare object vs object** | **Compare file vs file** |
| **Non-comparable** object | Display output   * Default (without ‘–v’ or’ –c’)   + display “Some objects are not comparable” * With ‘–v’ or ‘-c’   + Display reason of not-compatible; each type or space of given objects | Display output   * Default (without ‘-v’ or ‘-c’)   + display “Some objects are not comparable” * with ‘-v’ or ‘-c’   + Display reason of not-comparable; each type or space of not-comparable dataset |
| EXIT-CODE: 0 ([HDFFV-7628](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7628)) | EXIT-CODE: 0 ([HDFFV-7628](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7628)) |
|  | MISSING CODE: Identified a bug to compare different object types (dataset , group, type) as same name (common)   * Bug [HDFFV-7644](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7644) |
| **Non-comparable** attribute | Display output   * Default (without ‘-v’ or ‘-c’)   + display “Some objects are not comparable” * Use ‘-v’ or ‘-c’   + Display reason of not-comparable; each type or space of not-comparable attribute * Use ‘-v1’ or ‘-v2’ to view details   + attribute status line or list | Display output   * Default (without ‘-v’ or ‘-c’)   + display “Some objects are not comparable” * Use ‘-v’ or ‘-c’   + Display reason of not-comparable; each type or space of not-comparable attribute * Use ‘-v1’ or ‘-v2’ to view details   + attribute status line or list |
| EXIT-CODE: 0 ([HDFFV-7628](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7628)) | EXIT-CODE: 0 ([HDFFV-7628](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7628)) |
| ISSUE:   * Returning exit-code 1 would be sensible behavior. ([HDFFV-7628](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7628)) got previously entered to JIRA related to this behavior. * Missing code for comparing different object types as common object | | |

### 3.3.1 Example for non-comparable object

|  |  |
| --- | --- |
| **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| **$ h5diff obj-nocomparable1.h5 obj-nocomparable2.h5 /obj1**  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects.  **$ h5diff -c obj-nocomparable1.h5 obj-nocomparable2.h5 /obj1**  Not comparable: </obj1> is of class H5T\_INTEGER and </obj1> is of class H5T\_STRING  Not comparable: </obj1> has sign H5T\_SGN\_2 and </obj1> has sign H5T\_SGN\_ERROR  **$ h5diff -v obj-nocomparable1.h5 obj-nocomparable2.h5 /obj1**  dataset: </obj1> and </obj1>  Not comparable: </obj1> is of class H5T\_INTEGER and </obj1> is of class H5T\_STRING  Not comparable: </obj1> has sign H5T\_SGN\_2 and </obj1> has sign H5T\_SGN\_ERROR  0 differences found  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects. | **$ h5diff obj-nocomparable1.h5 obj-nocomparable2.h5**  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects.  **$ h5diff –c obj-nocomparable1.h5 obj-nocomparable2.h5**  Not comparable: </obj1> is of class H5T\_INTEGER and </obj1> is of class H5T\_STRING  Not comparable: </obj1> has sign H5T\_SGN\_2 and </obj1> has sign H5T\_SGN\_ERROR  **$ h5diff -v obj-nocomparable1.h5 obj-nocomparable2.h5**  file1 file2  ---------------------------------------  x x /  x x /obj1  x x /obj2  group : </> and </>  0 differences found  dataset: </obj1> and </obj1>  Not comparable: </obj1> is of class H5T\_INTEGER and </obj1> is of class H5T\_STRING  Not comparable: </obj1> has sign H5T\_SGN\_2 and </obj1> has sign H5T\_SGN\_ERROR  0 differences found  dataset: </obj2> and </obj2>  0 differences found  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects. |
| EXIT CODE: 0 | EXIT CODE: 0 |

### 3.3.2 Example for mixed type for common object (missing code)

|  |  |
| --- | --- |
| **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| **$ h5diff -v mix-nocomparable1.h5 mix-nocomparable2.h5 /obj1**  </obj1> is of type H5G\_GROUP and </obj1> is of type H5G\_DATASET  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects.  **$ h5diff -v mix-nocomparable1.h5 mix-nocomparable2.h5 /obj2**  </obj2> is of type H5G\_TYPE and </obj2> is of type H5G\_GROUP  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects.  **$ h5diff -v mix-nocomparable1.h5 mix-nocomparable2.h5 /obj3**  </obj3> is of type H5G\_DATASET and </obj3> is of type H5G\_TYPE  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects. | **$ h5diff -v mix-nocomparable1.h5 mix-nocomparable2.h5**  file1 file2  ---------------------------------------  x x /  x x /obj1  x x /obj2  x x /obj3  group : </> and </>  0 differences found  group : </obj1> and </obj1>  0 differences found  HDF5-DIAG: Error detected in HDF5 (1.9.85) thread 0:  #000: ../../hdf5/src/H5G.c line 452 in H5Gopen2(): unable to open group  major: Symbol table  minor: Can't open object  #001: ../../hdf5/src/H5G.c line 1010 in H5G\_open\_name(): not a group  major: Symbol table  minor: Inappropriate type  HDF5-DIAG: Error detected in HDF5 (1.9.85) thread 0:  #000: ../../hdf5/src/H5Tcommit.c line 597 in H5Topen2(): not a named datatype  major: Datatype  minor: Inappropriate type  dataset: </obj3> and </obj3>  Cannot open dataset </obj3>  0 differences found |
| EXIT CODE: 0 | EXIT CODE: 2 |

## Other behaviors

## Description

This section is for known issues that didn’t get covered by the other sections. This section may include some issues from other sections with a different point of view.

### Empty file vs. Non empty file

|  |  |
| --- | --- |
| **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| N/A | **$ h5diff empty\_file.h5 extra-obj1.h5**  <None> |
| EXIT-CODE : 1 |
|  | ISSUE: No message as default. This is same issue with the extra-object case |

# Appendix A – List of example HDF5 files (in alphabetic order)

|  |  |
| --- | --- |
| **danglelinks1.h5** | **danglelinks2.h5** |
| HDF5 "danglelinks1.h5" {  GROUP "/" {  EXTERNAL\_LINK "extlink1" {  TARGETFILE "danglelinks2.h5"  TARGETPATH "later"  }  EXTERNAL\_LINK "extlink2" {  TARGETFILE "not-yet.h5"  TARGETPATH "not-yet"  }  EXTERNAL\_LINK "extlink3" {  TARGETFILE "danglelinks2.h5"  TARGETPATH "/dset"  DATASET "/dset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): 0, 0  }  }  }  SOFTLINK "slink1" {  LINKTARGET "/not-yet"  }  SOFTLINK "slink2" {  LINKTARGET "/not-yet"  }  }  } | HDF5 "danglelinks2.h5" {  GROUP "/" {  DATASET "dset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): 0, 0  }  }  EXTERNAL\_LINK "extlink1" {  TARGETFILE "danglelinks1.h5"  TARGETPATH "later"  }  EXTERNAL\_LINK "extlink2" {  TARGETFILE "later.h5"  TARGETPATH "later"  }  EXTERNAL\_LINK "extlink3" {  TARGETFILE "later.h5"  TARGETPATH "later"  }  SOFTLINK "slink1" {  LINKTARGET "/later"  }  SOFTLINK "slink2" {  LINKTARGET "/dset"  }  }  } |

|  |  |
| --- | --- |
| **dset\_empty1.h5** | **dset\_empty2.h5** |
| HDF5 "dset\_empty1.h5" {  GROUP "/" {  DATASET "empty\_d1" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2, 2 ) / ( 2, 2 ) }  DATA {  (0,0): 0, 0,  (1,0): 0, 0  }  }  DATASET "empty\_d2" {  DATATYPE H5T\_IEEE\_F32LE  DATASPACE SIMPLE { ( 2, 2 ) / ( 2, 2 ) }  DATA {  (0,0): 0, 0,  (1,0): 0, 0  }  }  }  } | HDF5 "dset\_empty2.h5" {  GROUP "/" {  DATASET "empty\_d1" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2, 2 ) / ( 2, 2 ) }  DATA {  (0,0): 0, 0,  (1,0): 0, 0  }  }  DATASET "empty\_d2" {  DATATYPE H5T\_IEEE\_F32LE  DATASPACE SIMPLE { ( 2, 2 ) / ( 2, 2 ) }  DATA {  (0,0): 0, 0,  (1,0): 0, 0  }  }  }  } |

|  |  |
| --- | --- |
| **dset-num1\_1.h5** | **dset-num1\_2.h5** |
| HDF5 "dset-num1\_1.h5" {  GROUP "/" {  DATASET "dset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2, 2 ) / ( 2, 2 ) }  DATA {  (0,0): 1, 2,  (1,0): 3, 4  }  }  }  } | HDF5 "dset-num1\_2.h5" {  GROUP "/" {  DATASET "dset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2, 2 ) / ( 2, 2 ) }  DATA {  (0,0): 0, 0,  (1,0): 0, 0  }  }  }  } |

|  |  |
| --- | --- |
| **dset-num2\_1.h5** | **dset-num2\_2.h5** |
| HDF5 "dset-num2\_1.h5" {  GROUP "/" {  DATASET "dset1" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 4 ) / ( 4 ) }  DATA {  (0): 0, 0, 0, 0  }  }  }  } | HDF5 "dset-num2\_2.h5" {  GROUP "/" {  DATASET "dset1" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 4 ) / ( 4 ) }  DATA {  (0): 10, 20, 30, 40  }  }  }  } |

|  |  |
| --- | --- |
| **dset-num3\_1.h5** | **dset-num3\_2.h5** |
| HDF5 "dset-num3\_1.h5" {  GROUP "/" {  DATASET "dset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3, 2 ) / ( 3, 2 ) }  DATA {  (0,0): 100, 100,  (1,0): 100, 0,  (2,0): 0, 100  }  }  }  } | HDF5 "dset-num3\_2.h5" {  GROUP "/" {  DATASET "dset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3, 2 ) / ( 3, 2 ) }  DATA {  (0,0): 120, 80,  (1,0): 0, 100,  (2,0): 0, 50  }  }  }  } |

|  |  |
| --- | --- |
| **dset-space-array1.h5** | **dset-space-array2.h5** |
| HDF5 "dset-space-array1.h5" {  GROUP "/" {  DATASET "space-array" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2, 3 ) / ( 2, 3 ) }  DATA {  (0,0): 0, 0, 0,  (1,0): 1, 1, 1  }  }  }  } | HDF5 "dset-space-array2.h5" {  GROUP "/" {  DATASET "space-array" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2, 3 ) / ( 2, 3 ) }  DATA {  (0,0): 0, 10, 0,  (1,0): 10, 1, 10  }  }  }  } |

|  |  |
| --- | --- |
| **dset-str1\_1.h5** | **dset-str1\_2.h5** |
| HDF5 "dset-str1\_1.h5" {  GROUP "/" {  DATASET "string" {  DATATYPE H5T\_STRING {  STRSIZE 3;  STRPAD H5T\_STR\_NULLTERM;  CSET H5T\_CSET\_ASCII;  CTYPE H5T\_C\_S1;  }  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): "yyy", "zzz"  }  }  }  } | HDF5 "dset-str1\_2.h5" {  GROUP "/" {  DATASET "string" {  DATATYPE H5T\_STRING {  STRSIZE 3;  STRPAD H5T\_STR\_NULLTERM;  CSET H5T\_CSET\_ASCII;  CTYPE H5T\_C\_S1;  }  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): "cat", "dog"  }  }  }  } |

|  |  |
| --- | --- |
| **dset-str2\_1.h5** | **dset-str2\_2.h5** |
| HDF5 "dset-str2\_1.h5" {  GROUP "/" {  DATASET "note" {  DATATYPE H5T\_STRING {  STRSIZE 13;  STRPAD H5T\_STR\_NULLPAD;  CSET H5T\_CSET\_ASCII;  CTYPE H5T\_C\_S1;  }  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): "This is a dog", "This is a dog", "This is a dog"  }  }  }  } | HDF5 "dset-str2\_2.h5" {  GROUP "/" {  DATASET "note" {  DATATYPE H5T\_STRING {  STRSIZE 13;  STRPAD H5T\_STR\_NULLPAD;  CSET H5T\_CSET\_ASCII;  CTYPE H5T\_C\_S1;  }  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): "This is a dog", "This is a cat", "This is a dog"  }  }  }  } |

|  |  |
| --- | --- |
| **dset-type-array1.h5** | **dset-type-array2.h5** |
| HDF5 "dset-type-array1.h5" {  GROUP "/" {  DATASET "type-array" {  DATATYPE H5T\_ARRAY { [3] H5T\_STD\_I32LE }  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): [ 0, 0, 0 ], [ 1, 1, 1 ]  }  }  }  } | HDF5 "dset-type-array2.h5" {  GROUP "/" {  DATASET "type-array" {  DATATYPE H5T\_ARRAY { [3] H5T\_STD\_I32LE }  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): [ 0, 10, 0 ], [ 10, 1, 10 ]  }  }  }  } |

|  |
| --- |
| **empty\_file.h5** |
| HDF5 "empty\_file.h5" {  GROUP "/" {  }  } |

|  |
| --- |
| **enum\_invalid.h5** |
| HDF5 "enum\_invalid.h5" {  GROUP "/" {  DATASET "dset1" {  DATATYPE H5T\_ENUM {  H5T\_STD\_I32LE;  "YIN" 0;  "YANG" 1;  }  DATASPACE SIMPLE { ( 6 ) / ( 6 ) }  DATA {  (0): 09:00:00:00, YIN, 09:00:00:00, YIN, 09:00:00:00, YIN  }  }  DATASET "dset2" {  DATATYPE H5T\_ENUM {  H5T\_STD\_I32LE;  "YIN" 0;  "YANG" 1;  }  DATASPACE SIMPLE { ( 6 ) / ( 6 ) }  DATA {  (0): 09:00:00:00, 09:00:00:00, YIN, YIN, 09:00:00:00, YANG  }  }  }  } |

|  |  |
| --- | --- |
| **extra-attr1.h5** | **extra-attr2.h5** |
| HDF5 "extra-attr1.h5" {  GROUP "/" {  DATASET "do" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): 1, 1  }  ATTRIBUTE "attr1" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 1 ) / ( 1 ) }  DATA {  (0): 1  }  }  }  }  } | HDF5 "extra-attr2.h5" {  GROUP "/" {  DATASET "do" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): 1, 1  }  ATTRIBUTE "attr1" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 1 ) / ( 1 ) }  DATA {  (0): 1  }  }  ATTRIBUTE "attr2" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 1 ) / ( 1 ) }  DATA {  (0): 2  }  }  }  }  } |
|  | The object “do” has extra attribute “attr2” |

|  |  |
| --- | --- |
| **extra-obj1.h5** | **extra-obj2.h5** |
| HDF5 "extra-obj1.h5" {  GROUP "/" {  DATASET "do" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): 1, 1  }  }  }  } | HDF5 "extra-obj2.h5" {  GROUP "/" {  DATASET "do" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): 1, 1  }  }  GROUP "mi" {  }  DATASET "re" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 1, 2 ) / ( 1, 2 ) }  DATA {  (0,0): 3, 3  }  }  }  } |
|  | This file has extra object dataset “re” and group “mi”. |

|  |  |
| --- | --- |
| **groups1.h5** | **groups2.h5** |
| HDF5 "groups1.h5" {  GROUP "/" {  GROUP "grp1" {  DATASET "dset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): 1, 0, 0  }  }  GROUP "grp2" {  DATASET "dset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): 2, 0, 0  }  }  }  }  }  } | HDF5 "groups2.h5" {  GROUP "/" {  GROUP "grp1" {  DATASET "dset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): 1, 1, 0  }  }  GROUP "grp2" {  DATASET "dset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): 2, 2, 0  }  }  }  }  }  } |

|  |  |
| --- | --- |
| **mix-nocomparable1.h5** | **mix-nocomparable2.h5** |
| HDF5 "mix-nocomparable1.h5" {  GROUP "/" {  GROUP "obj1" {  }  DATATYPE "obj2" H5T\_STD\_I32LE;  DATASET "obj3" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2, 1 ) / ( 2, 1 ) }  DATA {  (0,0): 1,  (1,0): 0  }  }  }  } | HDF5 "mix-nocomparable2.h5" {  GROUP "/" {  DATASET "obj1" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2, 2 ) / ( 2, 2 ) }  DATA {  (0,0): 0, 1,  (1,0): 1, 0  }  }  GROUP "obj2" {  }  DATATYPE "obj3" H5T\_STD\_I32LE;  }  } |

|  |  |
| --- | --- |
| **obj-nocomparable1.h5** | **obj-nocomparable2.h5** |
| HDF5 "obj-nocomparable1.h5" {  GROUP "/" {  DATASET "obj1" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): 1, 2  }  }  DATASET "obj2" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 1, 1 ) / ( 1, 1 ) }  DATA {  (0,0): 10  }  }  }  } | HDF5 "obj-nocomparable2.h5" {  GROUP "/" {  DATASET "obj1" {  DATATYPE H5T\_STRING {  STRSIZE 5;  STRPAD H5T\_STR\_NULLPAD;  CSET H5T\_CSET\_ASCII;  CTYPE H5T\_C\_S1;  }  DATASPACE SIMPLE { ( 1, 1 ) / ( 1, 1 ) }  DATA {  (0,0): "abcde"  }  }  DATASET "obj2" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 1, 1 ) / ( 1, 1 ) }  DATA {  (0,0): 10  }  }  }  } |

|  |  |
| --- | --- |
| **ref-dsetreg1.h5** | **ref-dsetreg2.h5** |
| HDF5 "ref-dsetreg1.h5" {  GROUP "/" {  DATASET "Dset\_REGREF" {  DATATYPE H5T\_REFERENCE { H5T\_STD\_REF\_DSETREG }  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  DATASET /dset {(0,1), (2,11), (1,0), (2,4)},  DATASET /dset {(0,0)-(0,2), (0,11)-(0,13), (2,0)-(2,2), (2,11)-(2,13)}  }  }  DATASET "dset" {  DATATYPE H5T\_STD\_I8LE  DATASPACE SIMPLE { ( 3, 16 ) / ( 3, 16 ) }  DATA {  (0,0): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  (1,0): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  (2,0): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0  }  }  }  } | HDF5 "ref-dsetreg2.h5" {  GROUP "/" {  DATASET "Dset\_REGREF" {  DATATYPE H5T\_REFERENCE { H5T\_STD\_REF\_DSETREG }  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  DATASET /dset {(0,1), (2,11), (1,0), (2,4)},  DATASET /dset {(0,0)-(0,2), (0,11)-(0,13), (2,0)-(2,2), (2,11)-(2,13)}  }  }  DATASET "dset" {  DATATYPE H5T\_STD\_I8LE  DATASPACE SIMPLE { ( 3, 16 ) / ( 3, 16 ) }  DATA {  (0,0): 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  (1,0): 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  (2,0): 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3  }  }  }  } |

|  |  |
| --- | --- |
| **ref-obj1.h5** | **ref-obj2.h5** |
| HDF5 "ref-obj1.h5" {  GROUP "/" {  DATASET "Dset1" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): 0, 0, 0  }  }  DATASET "Dset\_OBJREF" {  DATATYPE H5T\_REFERENCE { H5T\_STD\_REF\_OBJECT }  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): DATASET 800 /Dset1 , GROUP 1400 /Group ,  (2): DATATYPE 2104 /NamedDatatype  }  }  GROUP "Group" {  DATASET "Dset2" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): 0, 0, 0  }  }  }  DATATYPE "NamedDatatype" H5T\_STD\_I32LE;  }  } | HDF5 "ref-obj2.h5" {  GROUP "/" {  DATASET "Dset1" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): 1, 0, 0  }  }  DATASET "Dset\_OBJREF" {  DATATYPE H5T\_REFERENCE { H5T\_STD\_REF\_OBJECT }  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): DATASET 800 /Dset1 , GROUP 1400 /Group ,  (2): DATATYPE 2104 /NamedDatatype  }  }  GROUP "Group" {  DATASET "Dset2" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): 0, 0, 2  }  }  }  DATATYPE "NamedDatatype" H5T\_STD\_I8LE;  }  } |

|  |  |
| --- | --- |
| **softlinks1.h5** | **softlinks2.h5** |
| HDF5 "softlinks1.h5" {  GROUP "/" {  DATASET "dset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): 1, 0, 0  }  }  GROUP "grp" {  DATASET "gdset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): 1, 1, 0  }  }  }  SOFTLINK "softlink2dset" {  LINKTARGET "/dset"  }  SOFTLINK "softlink2grp" {  LINKTARGET "/grp"  }  }  } | HDF5 "softlinks2.h5" {  GROUP "/" {  DATASET "dset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): 2, 0, 0  }  }  GROUP "grp" {  DATASET "gdset" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 3 ) / ( 3 ) }  DATA {  (0): 1, 2, 0  }  }  }  SOFTLINK "softlink2dset" {  LINKTARGET "/dset"  }  SOFTLINK "softlink2grp" {  LINKTARGET "/grp"  }  }  } |

# Acknowledgements

This work is supported by a commercial client of the HDF group.

# Revision History

|  |  |
| --- | --- |
| *August 11, 2011:* | Version 1 draft 1 circulated for directional comment within The HDF Group. |
| *August 24, 2011:* | Version 1 draft 2 circulated for directional comment within The HDF Group |