**The h5diff’s current behaviors and shortcomings**

Jonathan Kim

# Purpose

Examine nonsensical behaviors and potential improvements from the list of h5diff’s current behaviors.

With the results, we can prevent similar issues from the new ***h5compare*** tool, and also decide what to do with current ***h5diff***.

This document can be used for a reference for developing the new comparing tool; h5compare.

This document also can be evolved to be a behavioral example reference for h5diff once decisions are made.

# 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) As a result, the current h5diff tool implements some incorrect and inconsistent behaviors which confuse users.

# Main chapters

There are four main chapters to show the current h5diff tool’s behaviors.

1. Default behaviors - h5diff’s behavior without using options
2. **Behavior with options** - additional or exceptional behavior with using options
3. How to handle common, unique or non-comparable object and attribute with results - what user would look and feel in a big picture
4. Other behaviors - other known issues

Overview of Contents

[1. Default behaviors 7](#_Toc325616088)

[1.1 Dataset : *Number* 7](#_Toc325616089)

[1.2 Dataset : *String* 8](#_Toc325616090)

[1.3 Dataset: Array (integer type) 9](#_Toc325616091)

[1.4 Dataset: Array (ARRAY type) 10](#_Toc325616092)

[1.5 Dataset: Object Reference 11](#_Toc325616093)

[1.6 Dataset: Region Reference 14](#_Toc325616094)

[1.7 Dataset: *Empty* 15](#_Toc325616095)

[1.8 Dataset: Invalid enum value 16](#_Toc325616096)

[1.9 Group 17](#_Toc325616097)

[1.10 Named Datatype 19](#_Toc325616098)

[1.11 Dataset: Variable Length Data 20](#_Toc325616099)

[1.12 Dataset: Bitfields 22](#_Toc325616100)

[1.13 Dataset: Opaque 23](#_Toc325616101)

[1.14 Dataset: Compound 25](#_Toc325616102)

[2. Behavior with options 26](#_Toc325616103)

[2.1 ‘-n C’ or ‘--count=C’ 26](#_Toc325616104)

[2.2 ‘-d D’ or ‘--delta=D’ 27](#_Toc325616105)

[2.3 ‘-p R’ or ‘–relative=R’ 29](#_Toc325616106)

[2.4 --use-system-epsilon 31](#_Toc325616107)

[2.5 --exclude-path “path” 31](#_Toc325616108)

[2.6 --follow-symlinks 32](#_Toc325616109)

[2.7 --no-dangling-links 34](#_Toc325616110)

[2.8 ‘-N’ or ‘--nan’ 36](#_Toc325616111)

[2.9 ‘-v1’ and ‘-v2’ 38](#_Toc325616112)

[2.10 ‘-c’ or ‘--compare’ 38](#_Toc325616113)

[3. Handle common, unique or non-comparable object and attribute with results 39](#_Toc325616114)

[3.1 When difference found in Common object or attribute 40](#_Toc325616117)

[3.2 When difference is due to Unique object or attribute 40](#_Toc325616118)

[3.3 When difference is due to Non-comparable object or attribute 43](#_Toc325616121)

[4. Other behaviors 46](#_Toc325616124)

[4.1 Compare Empty file vs. Non empty file 46](#_Toc325616126)

[4.2 Different behaviors between dataset and attribute when type-sign difference exist 47](#_Toc325616127)

[Appendix A – List of example HDF5 files (in alphabetic order) 50](#_Toc325616130)

# Content Format

Describing h5diff’s behaviors often caused more confusion, especially when the behavior is insensible.

So this format is put together to show each behavior, as if viewer types the command to see the output directly. It’s WYSWYG (What You See What You Get) concept to reduce misunderstanding or misinterpretation.

Throughout the rest pages, the current behavior of the h5diff will be shown along with exact output and exit-code for each behavior to be examined.

# **Highlights**

There are two styles of highlights with different colors which Indicate examining point to discuss for a decision.

* ISSUE – insensible behavior.
* IMPROVE – improvement to consider.
* SUMMARY – Decision or feedback from discussions.

# Example HDF5 files

All the HDF5 files that are used in each section can be found in “Appendix A”.

The files are available to run actual test.

# Default behaviors

This section shows h5diff’s behaviors without using options. However ‘-v’ option is used to show details for verification.

There are two ways to run h5diff command.

One way is only to compare the two objects; other way is to compare the two HDF5 files.

Both ways will be shown side by side in the first section 1.1 as example. However only one way will be used for the most of cases to make easy on eyes since both ways show similar result for the purpose.

## Dataset : *Number*

Shows how h5diff compares number data in dataset.

|  |  |
| --- | --- |
| **Compare *Object* vs *Object*** | **Compare *File* vs *File*** |
| $ **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 |
| EXIT-CODE : 1 | EXIT-CODE : 1 |

#### Show details with ‘-v’ option

|  |  |
| --- | --- |
| $ **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 |

|  |
| --- |
| SUMMARY  POSSIBLE IMPROVE (for h5compare)   * E: for file difference output – table showing is a bit difficult to understand without explanation. * D: more uniform output format would be good. |

## Dataset : *String*

Shows how h5diff compares string data in dataset.

|  |
| --- |
| $ **h5diff dset-str1\_1.h5 dset-str1\_2.h5 /string**  dataset: </string> and </string>  6 differences found |
| EXIT-CODE : 1 |

#### Show details with ‘-v’ option

|  |
| --- |
| $ **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 |

|  |
| --- |
| IMPROVE: 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” |

|  |
| --- |
| SUMMARY   * Agreed for comparing string as a unit * More info displaying would be helpful. Add offset info column. – refer to the SUMMARY in 1.4   h5compre   * Default as comparing string as a unit. * Option feature for comparing by each character. * Displaying control for showing entire string vs. only different portion of the string.   h5diff   * Ask our customers if they want the feature with an option. |

## Dataset: Array (integer type)

Shows how h5diff compares number data in dataset of integer type array.

|  |
| --- |
| **$ h5diff dset-array1.h5 dset-array2.h5 /array**  dataset: </array> and </array>  3 differences found |
| EXIT-CODE : 1 |

#### Show details with ‘-v’ option

|  |
| --- |
| **$ h5diff -v dset-array1.h5 dset-array2.h5 /array**  dataset: </array> and </array>  size: [2x3] [2x3]  position array array difference  ---------------------------------------------------------  [ 0 1 ] 0 10 10  [ 1 0 ] 1 10 9  [ 1 2 ] 1 10 9  3 differences found |

## Dataset: Array (ARRAY type)

Shows how h5diff compares number data in dataset of ARRAY type integer array.

|  |
| --- |
| **$ h5diff dset-type-array1.h5 dset-type-array2.h5 /type-array**  dataset: </type-array> and </type-array>  3 differences found |
| EXIT-CODE : 1 |

#### Show details with ‘-v’ option

|  |
| --- |
| **$ 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 |

|  |
| --- |
| IMPROVE: 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. |

|  |
| --- |
| SUMMARY   * Agreed for comparing Array type element as a unit. * Qin: need to show more info. Array offset – sub index. Add offset info column.   h5compre   * Default as comparing Array type element as a unit. * Optional feature for comparing each value in Array type element. * Displaying control for showing entire Array type element vs. only different portion of the element.   h5diff   * Ask our customers if they want the feature added with an option. |

## Dataset: Object Reference

Shows how h5diff compares data via Object Reference in dataset.

|  |  |
| --- | --- |
| **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. |
| EXIT-CODE : 1 | EXIT-CODE : 1 |
|  | ISSUE: Didn’t display difference for /Group/Dset2. However the difference is displayed with -v. |

#### Show details with ‘-c’ option

|  |  |
| --- | --- |
| **$ 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. | |

#### Show details with ‘-v’ option

|  |
| --- |
| **$ 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. |

|  |
| --- |
| ISSUE: Tthe 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. |

|  |
| --- |
| SUMMARY   * Agreed on the ‘ISSUE’   BUG:   * object reference to a group didn’t work – should be comparable * object reference to a Named-datatype didn’t work – should be comparable   OTHER ISSUE NEED DECISION:   * Need to decide follow or not in default? (This definition need to be added to higher level comparison document.) * h5diff   + Currently symbolic links doesn’t follow as default and follow with an option. However, as shown the above, h5diff follows an object reference to a dataset. So it needs to be discussed.   + E: Don’t follow as default and follow with an option. (the current symlink example) * h5compare   + Need to decide either act like h5diff or follow always as default? |

## Dataset: Region Reference

Shows how h5diff compares data via Region Reference in dataset.

|  |
| --- |
| **$ h5diff ref-dsetreg1.h5 ref-dsetreg2.h5 /Dset\_REGREF**  <NONE> |
| EXIT-CODE : 0 |

#### Show details with ‘-v’ option

|  |
| --- |
| **$ 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 |

|  |
| --- |
| ISSUE: Didn’t follow to compare region reference’s end point values. |

|  |
| --- |
| SUMMARY   * ‘ISSUE’ was viewed as expected. – not follow as default. * Consider adding following option.   NEED DECISION:   * However the decision for ‘following as default or not’ need to be made for h5diff and h5compare? (same issue with the 1.5) |

## Dataset: *Empty*

Shows how h5diff handles empty dataset.

‘Empty dataset’ is when the dataset’s storage size is 0.

|  |
| --- |
| **$ h5diff dset\_empty1.h5 dset\_empty2.h5 /empty\_d1**  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects. |
| EXIT-CODE : 0 |

#### Show details with ‘-c’ option

|  |
| --- |
| **$ h5diff -c dset\_empty1.h5 dset\_empty2.h5 /empty\_d1**  Not comparable: </empty\_d1> or </empty\_d1> is an empty dataset |

|  |
| --- |
| ISSUE: This empty dataset has same type and space. If type and space is same, empty should not be treated as non-comparable. It should be viewed as same dataset with empty condition. |

|  |
| --- |
| SUMMARY   * Agreed on the ‘ISSUE’ – should be same if both empty dataset has same type and space. * For both h5diff and h5compare.   FURTHER DISCUSSION about fill value comparison   * h5diff – doesn’t compare fill value * h5compre – consider comparing fill value for both ‘fill value vs. fill value’ and ‘fill value vs. real value’ cases. * For comparing ‘dataset with fill value’ vs. ‘dataset with real value’; storage status (allocated space vs. non-allocated space) can be by passed as it’s viewed as same dataset according to h5dump output, which would be also user’s view point. If it’s needed to differentiate them, provide it as an optional feature. |

## Dataset: Invalid enum value

Shows how h5diff handles invalid emun value in dataset.

Invalid enum value is a number which is not defined in the enum structure.

|  |
| --- |
| **$ h5diff enum\_invalid.h5 enum\_invalid.h5 /dset1 /dset2**  dataset: </dset1> and </dset2>  3 differences found |
| EXIT-CODE : 1 |

#### Show details with ‘-v’ option

|  |
| --- |
| **$ 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 |

|  |
| --- |
| IMPROVE: There is two pair of invalid enum values. We may consider displaying 5 differences. So user can aware the status of invalid values.  Refer to “enum\_invalid.h5” in APPENDIX A  ISSUE: how do we want to compare valid enum type? By only string, only assigned number? Or both? |

|  |
| --- |
| SUMMARY  DECISION   * h5diff - compare by string as default. * h5compare - compare by string as default. * Comparing by defined number or both number and string can be provided as optional feature if necessary.   NOTE:   * Not sure what’s decided for the IMPROVE comment. Question: Is invalid vs. invalid enum same or different? |

## Group

Shows how h5diff compares two groups which contain objects. Recursive by default.

|  |
| --- |
| **$ 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 |
| EXIT-CODE : 1 |

|  |
| --- |
| IMPROVE: Blank line between objects would be easier to read the output. (should consider for h5compare with different level output) |

|  |
| --- |
| SUMMARY   * Agreed on the ‘IMPROVE’ – add blank line in between objects. * h5diff – not support (leave as is). * h5compare – support as default.   OTHER IMPROVE:   * An optional feature for shallow comparison for group. Only compare the top level without recursive. |

## Named Datatype

Shows how h5diff compares two Named datatypes.

|  |
| --- |
| **$ h5diff ref-obj1.h5 ref-obj2.h5 /NamedDatatype /NamedDatatype**  datatype: </NamedDatatype> and </NamedDatatype>  <no output of difference found> |
| EXIT-CODE : 1 |

#### Show details with ‘-v’ option

|  |
| --- |
| **$ h5diff -v ref-obj1.h5 ref-obj2.h5 /NamedDatatype /NamedDatatype**  datatype: </NamedDatatype> and </NamedDatatype>  1 differences found |

|  |
| --- |
| ISSUE: Should display “1 differences found” also for default. |

|  |
| --- |
| SUMMARY  DECISION:   * It’s a BUG. Should be treated as non-comparable since the target types are different. ( We decided that ‘Named-datatype’ follows hard link comparison convention, which is always following to the target value) |

## Dataset: Variable Length Data

Shows how h5diff handles variable length data.

#### Expected behavior

|  |
| --- |
| **$ h5diff -v vlendata1.h5 vlendata2.h5 /dset1**  dataset: </dset1> and </dset1>  size: [2] [2]  position dset1 dset1 difference  ------------------------------------------------------------  [ 1 ] 5 10 5  [ 1 ] 5 11 6  2 differences found |
| EXIT-CODE : 1 |

#### Issue case1.1

|  |
| --- |
| **$ h5diff -v vlendata1.h5 vlendata2.h5 /dset2**  dataset: </dset2> and </dset2>  0 differences found |
| EXIT-CODE : 0 |

#### Issue case1.2

|  |
| --- |
| **$ h5diff -v vlendata2.h5 vlendata1.h5 /dset2**  dataset: </dset2> and </dset2>  size: [2] [2]  position dset2 dset2 difference  ------------------------------------------------------------  [ 0 ] 1 0 1  [ 1 ] 12 0 12  2 differences found |
| EXIT-CODE : 1 |

|  |
| --- |
| ISSUE1: The above issue case1.1 and case1.2 shows outputs from comparing the “/dset2” in the same pair of HDF5 files. However results are different by swapping the two HDF5 files. The outputs should be same. Need to decide which output is correct.  ISSUE2: The “/dset2” contain different length of vlen data. It’s seems to be decided how to compare such case. |

## Dataset: Bitfields

Shows how h5diff handles bitfield data.

#### Case1 - expected behavior

|  |
| --- |
| **$ h5diff -v bitfields1.h5 bitfields2.h5 /bitfield\_8**  dataset: </bitfield\_8> and </bitfield\_8>  size: [4] [4]  position bitfield\_8 bitfield\_8 difference  ------------------------------------------------------------  [ 0 ] 255 15 240  [ 3 ] 252 240 12  2 differences found |
| EXIT-CODE : 1 |

#### Case2

|  |
| --- |
| **$ h5diff -v bitfields1.h5 bitfields2.h5 /bitfield\_16**  dataset: </bitfield\_16> and </bitfield\_16>  size: [2] [2]  position bitfield\_16 bitfield\_16 difference  ------------------------------------------------------------  [ 0 ] 255 15 240  [ 1 ] 252 240 12  2 differences found |
| EXIT-CODE : 1 |

|  |
| --- |
| IMPROVE: For the case2, output is same as ‘bitfield\_8’ case, need to be improved. |

#### Possible new output1 (Recommend)

|  |
| --- |
| $ h5diff -v bitfields1.h5 bitfields2.h5 /bitfield\_16  dataset: </bitfield\_16> and </bitfield\_16>  size: [2] [2]  position bitfield\_16 bitfield\_16 difference  ------------------------------------------------------------  [ 0 ] 0xff 0x0f 0xf0 (byte-1)  [ 1 ] 0xfc 0xf0 0x0c (byte-0)  2 differences found |
| Note: similar to the current way, just adding indication for user to figure out more. less change is needed. |

#### Possible new output2

|  |
| --- |
| $ h5diff -v bitfields1.h5 bitfields2.h5 /bitfield\_16  dataset: </bitfield\_16> and </bitfield\_16>  size: [2] [2]  position bitfield\_16 bitfield\_16 difference  ------------------------------------------------------------  [ 0 ] **ff**:fe **0f**:fe f0:00  [ 1 ] fd:**fc** fd:**f0** 00:0c  2 differences found |
| Issue: an element can be too long to fit in a COL |

#### Possible new output3

|  |
| --- |
| $ h5diff -v bitfields1.h5 bitfields2.h5 /bitfield\_16  dataset: </bitfield\_16> and </bitfield\_16>  size: [2] [2]  position bitfield\_16 bitfield\_16 difference  ------------------------------------------------------------  [ 0 ] 65534 4094 61440  [ 1 ] 65020 65008 12  2 differences found |
| Issue: a number can be too big to calculate |

## Dataset: Opaque

Shows how h5diff handles opaque data.

|  |
| --- |
| **$ h5diff -v**  **opaque1.h5 opaque2.h5 /opaque\_dset**  dataset: </opaque\_dset> and </opaque\_dset>  size: [1] [1]  position opaque\_dset opaque\_dset difference  ------------------------------------------------------------  [ 0 ] 10 11 1  [ 0 ] 10 12 2  2 differences found |
| EXIT-CODE : 1 |

|  |
| --- |
| ISSUE: Should it be “1 difference found”  IMPROVE: Improve output to be more intuitive and indicates where the differences are in the element? |

#### Possible new output1 (Recommend)

|  |
| --- |
| dataset: </opaque\_dset> and </opaque\_dset>  size: [1] [1]  position opaque\_dset opaque\_dset difference  ------------------------------------------------------------  [ 0 ] 0x0a 0x0c 2 (byte-1)  [ 0 ] 0x0a 0x0b 1 (byte-3)  1 differences found |
| Note: similar to the current way, just adding indication for user to figure out more. less change is needed. |

#### Possible new output2

|  |
| --- |
| dataset: </opaque\_dset> and </opaque\_dset>  size: [1] [1]  position opaque\_dset opaque\_dset difference  ------------------------------------------------------------  [ 0 ] 0a:0a:0a:0a:0a 0a:0b:0a:0c:0a 00:01:00:02:00  1 differences found |
| Issue: an element can be too long to fit in a COL |

#### Possible new output3

|  |
| --- |
| dataset: </opaque\_dset> and </opaque\_dset>  size: [1] [1]  position opaque\_dset opaque\_dset difference  ------------------------------------------------------------  [ 0 ] <Number1> <Number2> <Difference>  1 differences found |
| Issue: number can be too big to calculate |

## Dataset: Compound

Shows how h5diff handles compound data.

|  |
| --- |
| **$ h5diff -v**  **compound1.h5 compound2.h5**  file1 file2  ---------------------------------------  x x /  x x /dset  group : </> and </>  0 differences found  dataset: </dset> and </dset>  size: [2] [2]  position dset dset difference  ------------------------------------------------------------  [ 1 ] c X  [ 1 ] 1 3 2  [ 1 ] 0.5 0.7 0.2  3 differences found |
| EXIT-CODE : 1 |

# Behavior with options

Additional or exceptional behaviors by using options.

Beside of the entitled option, verbose options (ex: ‘-v’, ‘-c’) are also used to show details for verification.

## ‘-n C’ or ‘--count=C’

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

#### without ‘-n’

|  |
| --- |
| **$ 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 |

#### with ‘-n’

|  |
| --- |
| **$ 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 |

|  |
| --- |
| SUMMARY  OTHER IMPROVE:   * h5diff: leave as is * h5compare: Print more info indicates the status. Format as “first 2 differences shown out of 4 found” at the end. |

## ‘-d D’ or ‘--delta=D’

Display output if difference from the two numbers is bigger than number ‘D’ which is specified by user.

Print difference if (|a-b| > D). D must be a positive number.

#### without ‘-d’

|  |
| --- |
| **$ 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 |

#### with ‘-d’

|  |
| --- |
| **$ 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 |

|  |
| --- |
| SUMMARY  OTHER IMPROVE:   * h5diff: leave as is * h5compare: Print more info indicates the status. Format as “2 differences greater than 20 found.” at the end. |

## ‘-p R’ or ‘–relative=R’

Print difference if (|(a-b)/b| > R). R must be a positive number. Number ‘a’ and ‘b’ is resorted before calculation.

#### without ‘-p’

|  |
| --- |
| **$ 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 |

#### 

#### with ‘-p’

|  |
| --- |
| **$ 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 |
| All EXIT CODE: 1 |

|  |
| --- |
| IMPROVE: Divided by 0 displays “not comparable” in relative column. Is the word “not comparable” correct? How about “N/A” ? |

|  |
| --- |
| SUMMARY   * Agreed on the ‘IMPROVE’ – Display “Inf” instead of “not comparable” for both h5diff and h5compare.   BACKGROUND:   * Elena suggested looking for scientific TERM for divided by 0 case. Dana looked for it and suggested for “inf” as it’s infinite. |

## --use-system-epsilon

Same concept as option ‘-d D’. Only difference is using a ‘system defined value’ or a ‘pre-defined value’ instead of user specifying value.

: Print difference if (|a-b| > EPSILON), EPSILON is system defined value.

If the system epsilon is not defined, one of the following predefined values will be used:

FLT\_EPSILON = 1.19209E-07 for floating-point type

DBL\_EPSILON = 2.22045E-16 for double precision type

## --exclude-path “path”

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

#### without ‘--exclude-path’

|  |
| --- |
| **$ 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 |

#### with ‘--exclude-path’

|  |
| --- |
| **$ h5diff -v --exclude-path "/grp1/grp2" groups1.h5 groups2.h5**  file1 file2  ---------------------------------------  x x /  x x /grp1  x x /grp1/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 |
| EXIT CODE: 1 |

## --follow-symlinks

Follow symbolic links (soft links and external links) and compare the links' target objects. Refer to help page or RM for more details.

In this section ‘soft links’ are used to demonstrate the behavior.

#### without --follow-symlinks (soft link to a dataset )

|  |
| --- |
| **$ h5diff -v softlinks1.h5 softlinks2.h5 /softlink2dset**  link : </softlink2dset> and </softlink2dset>  0 differences found |
| EXIT CODE: 0 |

#### with --follow-symlinks ( soft link to a dataset )

|  |
| --- |
| **$ 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 |

#### without --follow-symlinks ( soft link to a group )

|  |
| --- |
| **$ h5diff -v softlinks1.h5 softlinks2.h5 /softlink2grp**  link : </softlink2grp> and </softlink2grp>  0 differences found |
| EXIT CODE: 0 |

#### with --follow-symlinks ( soft link to a group )

|  |
| --- |
| **$ 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 |

## --no-dangling-links

Treat dangling link as error. Must used with ‘--follow-symlinks’.

#### without ‘--no-dangling-links’

|  |
| --- |
| 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 |

#### with ‘--no-dangling-links’

|  |
| --- |
| **$ h5diff -v --follow-symlinks --no-dangling-links danglelinks1.h5** **danglelinks2.h5 /slink1**  Warning: </slink1> is a dangling link.  EXIT CODE: **2**  *NOTE: All the above cases are same. (display the first detected dangling link name)* |

#### only with ‘—follow-symlinks’

|  |
| --- |
| **$ h5diff -v --follow-symlinks danglelinks1.h5 danglelinks2.h5 /slink1**  obj1 </slink1> is a dangling link.  1 differences found  EXIT CODE: 1  *NOTE: All the above cases are same. (display the first detected dangling link name)* |

## ‘-N’ or ‘--nan’

With this option, ‘not a number’ vs. ‘not a number’ is treated as different.

In other hands, without this option (default), ‘not a number’ vs. ‘not a number’ is treated as same. To detect ‘not a number’ vs. ‘not a number’ is as same, it requires to perform multiple string comparisons because there are different string-symbols representing ‘not a number’. (ex: “NAN”, “nan”, “NaN”, "-1.#IND" an so on). However it takes longer time to compare all the possible ‘not a number’ strings as string comparison is slow.

This option is presented for users who don’t care to differentiate the two ‘not a numbers’ and want faster performance.

#### without --nan (care NaN)

|  |
| --- |
| **$ h5diff -v nans1.h5 nans2.h5**  file1 file2  ---------------------------------------  x x /  x x /d1  group : </> and </>  0 differences found  dataset: </d1> and </d1>  size: [5] [5]  position d1 d1 difference  ------------------------------------------------------------  [ 0 ] nan 0.2 nan  [ 1 ] 0.1 nan nan  [ 2 ] 0.1 0.3 0.2  3 differences found |
| EXIT CODE: 1 |

#### with --nan (not care NaN)

|  |
| --- |
| **$ h5diff -v --nan nans1.h5 nans2.h5**  file1 file2  ---------------------------------------  x x /  x x /d1  group : </> and </>  0 differences found  dataset: </d1> and </d1>  size: [5] [5]  position d1 d1 difference  ------------------------------------------------------------  [ 0 ] nan 0.2 nan  [ 1 ] 0.1 nan nan  [ 2 ] 0.1 0.3 0.2  [ 3 ] nan nan nan  4 differences found |
| EXIT CODE: 1 |

|  |
| --- |
| IMPROVE: Shouldn’t it be opposite way? So default should be not care NaN comparison. In the past ESDIS and Chicago customer needed to be told use –nan option if don’t care about NaN. |

|  |
| --- |
| SUMMARY  DECISION   * Leave it as is. (P: we already had changed back and forth a couple times)   OTHER IMPROVE:   * Display “- -“ instead of “nan” in the column of the difference because “nan” is confusing. * D: Use “--posix-nan” instead of “—nan” option. It gives more intuitive meaning for using the option. |

## ‘-v1’ and ‘-v2’

These options were added to display details of the attribute status. Refer to ‘unique attribute’ examples in section 3.

## ‘-c’ or ‘--compare’

List objects that are not comparable. Refer to ‘non-comparable’ examples in section3.

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

### Overview:

There are three common distinctions that a user must be aware of when making comparisons.

1. **When a difference is found in *common* object or attribute**
2. **When difference is due to *unique* object or attribute**
3. **When a difference is due to *non-comparable* object or attribute**

Terminology:

* **Common object**:
  + Comparing file vs. file; two absolute path names to an object are the same.
  + Comparing group vs. group; two relative path names (based in the specified groups) to an object are the same.
* **Common attribute**: Attribute names are the same when comparing the two objects.
* **Unique object**:
  + Comparing file vs. file, an absolute path name to an object exists only in one of the two files.
  + Comparing group vs. group, a relative path name (based in the specified group) to the object exists only in one of the two groups.
* **Unique attribute**: Attribute name exists only in one object when comparing two objects.
* **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’ or ‘file vs file’** |
| Display output   * Default (without ‘-v’ option)   + display number of differences with names of object or attribute * ‘–v’ option for details   + display details of the differences (data values)   + display number of differences with name of object or attribute |
| EXIT-CODE: 1 |

## When difference is due to Unique object or attribute

### 3.2.1 When unique object exists in any of the two files

#### Without option (default)

|  |
| --- |
| **$ h5diff unique-obj1.h5 unique-obj2.h5**  <no output> |
| EXIT-CODE : 1 |

#### With ‘-v’ to verify unique object

|  |
| --- |
| **$ h5diff -v unique-obj1.h5 unique-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 |

|  |
| --- |
| ISSUE: No output in default, which causes confusion to user. Some message should be displayed indicating there are differences in default. |

|  |
| --- |
| SUMMARY   * Improve to display “2 unique object found in <file2>” message for default (without option) and verbose mode.   h5diff   * Tool team will decide either improves or not. Refer to <[HDFFV-7921](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7921)>   h5compare :   * Follow the improvement displaying message. |

### 3.2.2 When unique attribute exists in any of the two objects

#### Without option (default)

|  |
| --- |
| **$ h5diff unique-attr1.h5 unique-attr2.h5 /do**  <No output>  EXIT-CODE : 0 |

|  |
| --- |
| ISSUE: No message output indicating the unique attribute(s) as default. Made user confused and caused inconvenience to figure out such differences, especially when there are many objects. |

#### With options: -v1 and –v2 to verify unique attribute

|  |
| --- |
| **$ h5diff -v1 unique-attr1.h5 unique-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  EXIT-CODE : 0 |
| **$ h5diff -v2 unique-attr1.h5 unique-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 |
| **$ h5diff -v unique-attr1.h5 unique-attr2.h5 /do**  dataset: </do> and </do>  attribute: <attr1 of </do>> and <attr1 of </do>>  0 differences found  0 differences found  EXIT-CODE : 0  <still can’t tell anything about unique attribute> |

|  |
| --- |
| ISSUE: The exit code 1 would be sensible behavior, which indicates the two objects are different. ([HDFFV-7643](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7643)) |

|  |
| --- |
| SUMMARY  h5diff   * Treat as a bug. . Return 1 for exit-code (HDFFV-7643)   h5compare   * Return 1 for exit-code.   Note   * Keep consistency with the above ‘unique object’ case (return exit code 1). * If need to return 0 for exit-code, provide it as an optional feature. |

## When difference is due to Non-comparable object or attribute

**You can refer to non-comparable documents at :** <http://confluence.hdfgroup.uiuc.edu/display/TOOLS/Non-Comparables>.

(However these issues are not addressed yet in any document.)

### Non-comparable object

#### 

#### Without option (default)

|  |
| --- |
| **$ h5diff obj-nocomparable1.h5 obj-nocomparable2.h5 /obj1**  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects.  EXIT CODE: 0 |

#### With options: -c and -v

|  |
| --- |
| **$ 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  EXIT CODE: 0 |
| **$ 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.  EXIT CODE: 0 |

|  |
| --- |
| ISSUE:   * Returning exit-code 1 would be sensible behavior. Turned out that this issue got previously entered as [HDFFV-7628](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7628) to JIRA by Elena.   BUG identified:   * For the ‘file vs file’ , identified a bug from missing code for comparing different object types as common object. ([HDFFV-7644](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7644)). |

|  |
| --- |
| SUMMARY  h5diff   * Treat as a bug. Return 1 for exit-code. (HDFFV-7628)   h5compare   * Return 1 for exit-code.   Note   * If need to return 0 for exit-code, provide it as an optional feature. |

### Non-comparable attribute

|  |
| --- |
| Display output  It’s same as shown in the above “Example for non-comparable object” for both default and optional. No issue with output display. |
| EXIT-CODE: 0 |

|  |
| --- |
| ISSUE:   * Returning exit-code 1 would be sensible behavior. **Same exit code issue as the above non-comparable object.** ([HDFFV-7628](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7628)) |

|  |
| --- |
| SUMMARY  h5diff   * Treat as a bug. Return 1 for exit-code. (HDFFV-7628)   h5compare   * Return 1 for exit-code.   Note   * Keep consistency with the above ‘non-comparable object’ case (return exit code 1). * If need to return 0 for exit-code, provide it as an optional feature. |

# 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.

## Compare Empty file vs. Non empty file

|  |
| --- |
| **$ h5diff empty\_file.h5 unique-obj1.h5**  <None> |
| EXIT-CODE : 1 |

|  |
| --- |
| ISSUE: No message as default. Some message should be displayed indicating the difference. This is same issue with the unique-object case |

|  |
| --- |
| SUMMARY  h5diff   * Tool team will discuss either improve or not.   h5compare   * Display some message according to the status of the difference. “Difference found: “empty\_file.h5 **“** is an empty file**.** |

#### Show details with ‘-v’ option

|  |
| --- |
| **$ h5diff -v empty\_file.h5 unique-obj1.h5**  file1 file2  ---------------------------------------  x x /  x /do  group : </> and </>  0 differences found |
| EXIT-CODE : 1 |

## Different behaviors between dataset and attribute when type-sign difference exist

### Sign difference in dataset

The sign difference is treated as non-comparable.

|  |
| --- |
| **$ h5diff -c dset\_attr\_sign1.h5 dset\_attr\_sign2.h5 /d1**  Not comparable: </d1> has sign H5T\_SGN\_2 and </d1> has sign H5T\_SGN\_NONE  EXIT-CODE : 0 |

#### Show details with ‘-v’ option

|  |
| --- |
| **$ h5diff -v dset\_attr\_sign1.h5 dset\_attr\_sign2.h5 /d1**  dataset: </d1> and </d1>  Warning: different storage datatype  </d1> has file datatype H5T\_STD\_I32LE  </d1> has file datatype H5T\_STD\_U32LE  Not comparable: </d1> has sign H5T\_SGN\_2 and </d1> has sign H5T\_SGN\_NONE  0 differences found  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects without details of differences.  EXIT-CODE : 0 |

### Sign difference in attribute

The sign difference is NOT treated as non-comparable.

|  |
| --- |
| **$ h5diff -c dset\_attr\_sign1.h5 dset\_attr\_sign2.h5 /g1**  attribute: <attr of </g1>> and <attr of </g1>>  2 differences found  EXIT-CODE : 1 |

#### Show details with ‘-v’ option

|  |
| --- |
| **$ h5diff -v dset\_attr\_sign1.h5 dset\_attr\_sign2.h5 /g1**  group1 group2  ---------------------------------------  x x  group : </g1> and </g1>  0 differences found  Warning: different storage datatype  <attr> has file datatype H5T\_STD\_I32LE  <attr> has file datatype H5T\_STD\_U32LE  attribute: <attr of </g1>> and <attr of </g1>>  size: [2] [2]  position attr of </g1> attr of </g1> difference  ------------------------------------------------------------  [ 0 ] 1 3 2  [ 1 ] 2 4 2  2 differences found  EXIT-CODE : 1 |

|  |
| --- |
| ISSUE:   * Different behaviors: when sign difference exists in dataset, it’s treated as non-comparable. When sign difference exists in attribute, it’s not treated as non-comparable. * This issue was entered in JIRA as [HDFFV-7725](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7725). Not sure which one is intended behavior. |

|  |
| --- |
| SUMMARY  h5diff   * Treat it as a bug. Dataset case is correct. (HDFFV-7725)   h5compare   * Whoever works on h5compare tool will need to come up with a decision with tool team. |

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

|  |  |
| --- | --- |
| **bitfields1.h5** | **bitfields2.h5** |
| HDF5 "bitfields1.h5" {  GROUP "/" {  DATASET "bitfield\_16" {  DATATYPE H5T\_STD\_B16LE  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): ff:fe, fd:fc  }  }  DATASET "bitfield\_8" {  DATATYPE H5T\_STD\_B8LE  DATASPACE SIMPLE { ( 4 ) / ( 4 ) }  DATA {  (0): 0xff, 0xfe, 0xfd, 0xfc  }  }  }  } | HDF5 "bitfields2.h5" {  GROUP "/" {  DATASET "bitfield\_16" {  DATATYPE H5T\_STD\_B16LE  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): 0f:fe, fd:f0  }  }  DATASET "bitfield\_8" {  DATATYPE H5T\_STD\_B8LE  DATASPACE SIMPLE { ( 4 ) / ( 4 ) }  DATA {  (0): 0x0f, 0xfe, 0xfd, 0xf0  }  }  }  } |

|  |  |
| --- | --- |
| **compound1.h5** | **compound2.h5** |
| HDF5 "compound1.h5" {  GROUP "/" {  DATASET "dset" {  DATATYPE H5T\_COMPOUND {  H5T\_STRING {  STRSIZE 6;  STRPAD H5T\_STR\_NULLTERM;  CSET H5T\_CSET\_ASCII;  CTYPE H5T\_C\_S1;  } "c\_name";  H5T\_STD\_I16LE "d\_name";  H5T\_IEEE\_F64LE "f\_name";  }  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): {  "abcdef",  0,  0.5  },  (1): {  "abcdef",  1,  0.5  }  }  }  }  } | HDF5 "compound2.h5" {  GROUP "/" {  DATASET "dset" {  DATATYPE H5T\_COMPOUND {  H5T\_STRING {  STRSIZE 6;  STRPAD H5T\_STR\_NULLTERM;  CSET H5T\_CSET\_ASCII;  CTYPE H5T\_C\_S1;  } "c\_name";  H5T\_STD\_I16LE "d\_name";  H5T\_IEEE\_F64LE "f\_name";  }  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): {  "abcdef",  0,  0.5  },  (1): {  "abXdef",  3,  0.7  }  }  }  }  } |

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

|  |  |
| --- | --- |
| **dset\_attr\_sign1.h5** | **dset\_attr\_sign2.h5** |
| HDF5 "dset\_attr\_sign1.h5" {  GROUP "/" {  DATASET "d1" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): 1, 2  }  }  GROUP "g1" {  ATTRIBUTE "attr" {  DATATYPE H5T\_STD\_I32LE  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): 1, 2  }  }  }  }  } | HDF5 "dset\_attr\_sign2.h5" {  GROUP "/" {  DATASET "d1" {  DATATYPE H5T\_STD\_U32LE  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): 3, 4  }  }  GROUP "g1" {  ATTRIBUTE "attr" {  DATATYPE H5T\_STD\_U32LE  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): 3, 4  }  }  }  }  } |

|  |  |
| --- | --- |
| **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-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  }  }  }  } |

|  |  |
| --- | --- |
| **unique-attr1.h5** | **unique-attr2.h5** |
| HDF5 "unique-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 "unique-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 unique attribute “attr2” |

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

|  |  |
| --- | --- |
| **nans1.h5** | **nans2.h5** |
| HDF5 "nans1.h5" {  GROUP "/" {  DATASET "d1" {  DATATYPE H5T\_IEEE\_F32LE  DATASPACE SIMPLE { ( 5 ) / ( 5 ) }  DATA {  (0): nan, 0.1, 0.1, nan, 0.1  }  }  }  } | HDF5 "nans2.h5" {  GROUP "/" {  DATASET "d1" {  DATATYPE H5T\_IEEE\_F32LE  DATASPACE SIMPLE { ( 5 ) / ( 5 ) }  DATA {  (0): 0.2, nan, 0.3, nan, 0.1  }  }  }  } |

|  |  |
| --- | --- |
| **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  }  }  }  } |

|  |  |
| --- | --- |
| **opaque1.h5** | **opaque2.h5** |
| HDF5 "opaque1.h5" {  GROUP "/" {  DATASET "opaque\_dset" {  DATATYPE H5T\_OPAQUE {  OPAQUE\_TAG "test opaque type - 2";  }  DATASPACE SIMPLE { ( 1 ) / ( 1 ) }  DATA {  (0): 0a:0a:0a:0a:0a  }  }  }  } | HDF5 "opaque2.h5" {  GROUP "/" {  DATASET "opaque\_dset" {  DATATYPE H5T\_OPAQUE {  OPAQUE\_TAG "test opaque type - 2";  }  DATASPACE SIMPLE { ( 1 ) / ( 1 ) }  DATA {  (0): 0a:0b:0a:0c:0a  }  }  }  } |

|  |  |
| --- | --- |
| **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"  }  }  } |

|  |  |
| --- | --- |
| **vlendata1.h5** | **vlendata2.h5** |
| HDF5 "vlendata1.h5" {  GROUP "/" {  DATASET "dset1" {  DATATYPE H5T\_VLEN { H5T\_STD\_I32LE}  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): (0), (5, 5)  }  }  DATASET "dset2" {  DATATYPE H5T\_VLEN { H5T\_STD\_I32LE}  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): (0), (10, 11)  }  }  }  } | HDF5 "vlendata2.h5" {  GROUP "/" {  DATASET "dset1" {  DATATYPE H5T\_VLEN { H5T\_STD\_I32LE}  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): (0), (10, 11)  }  }  DATASET "dset2" {  DATATYPE H5T\_VLEN { H5T\_STD\_I32LE}  DATASPACE SIMPLE { ( 2 ) / ( 2 ) }  DATA {  (0): (0, 1), (10, 11, 12)  }  }  }  } |

# 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. |
| *November 22, 2011:* | Version 1 draft 3 initially done. Reviewed and discussed with Peter. |
| *December 01, 2011:* | Version2 circulated for review and discussion within the HDF Group. |
| *January 20,2012:* | Version2.6 had meeting#1. Added ‘SUMMARY’ sections from the meeting. Minor updates. |
| *February 2,2012:* | Version2.7 circulated for the 2nd meeting within the HDF group |
| *February 07,2012:* | Version2.8 had meeting#2. Added ‘SUMMARY’ sections in chapter 3 and 4 from the meeting. Also made some updates. |
| *May 25,2012* | Version3.0 – add other datatypes (vlen, bitfields, opaque and compound) in ‘1. default behaviors’ section |