# The h5diff’s current behaviors and shortcomings

## Purpose

List h5diff’s current behaviors, so sensible and insensible behaviors can be figured out. With the result, we can decide what to do with current h5diff’s issues and also prevent those issues from the new h5compre tool.

## Background

Many features got implemented into the current h5diff command tool over the past. However some of the feature development procedures failed to address related updates which must go along with the feature (ex: default output or exit code and so on). Also lack of fundamental definitions contributed other insensible behaviors in a narrow scope of development procedures.

As a result, the current h5diff provides some inconsistent and insensible behaviors which confuse users to detect differences between HDF5 files.

In deciding developing a new comparing tool ‘h5ocompare’ for the future replacement, this document can be referred for comparing behaviors in sensible manner.

## Description

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

1. Default behaviors - main purposed 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. Insensible behaviors - known issues (this can grow as identifying more issues)

All of these will contain example output and exit-code from h5diff.

**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.

## Default behaviors

Assume all the objects are comparable object. (ex: same name, type, space)

|  |  |  |
| --- | --- | --- |
|  | **Compare *Object* vs *Object***  (ex: $h5diff file1.h5 file2.h5 /dset1 /dset2) | **Compare *File* vs *File***  (ex: $h5diff file1.h5 file2.h5) |
| Dataset : *Number* | $ **h5diff dset-num1\_1.h5 dset-num1\_2.h5 /dset /dset**  dataset: </dset> and </dset>  2 differences found | $ **h5diff dset-num1\_1.h5 dset-num1\_2.h5**  dataset: </dset> and </dset>  2 differences found |
| EXIT-CODE : 1 | EXIT-CODE : 1 |

|  |  |  |
| --- | --- | --- |
| Dataset :  *String* | $ **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 |
| EXIT-CODE : 1 | EXIT-CODE : 1 |
| NOTE: Refer to verbose output and consideration in section ‘2. Optional behaviors’ | |

|  |  |  |
| --- | --- | --- |
| Dataset:  *Space array* | **$ 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 |
| EXIT-CODE : 1 | EXIT-CODE : 1 |

|  |  |  |
| --- | --- | --- |
| Dataset:  *Type array* | **$ 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 |
| EXIT-CODE : 1 | EXIT-CODE : 1 |

## Continue (default behaviors)

|  |  |  |
| --- | --- | --- |
|  | **Compare *Object* vs *Object*** | **Compare *File* vs *File*** |
| Dataset:  *Obj reference* | **$ h5diff ref-obj1.h5 ref-obj2.h5 /Dset\_OBJREF**  --------------------------------  Some objects are not comparable  --------------------------------  Use -c for a list of objects.  **$/h5diff -c ref-obj1.h5 ref-obj2.h5 /Dset\_OBJREF**  <NONE> | **$ 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**  dataset: </Dset1> and </Dset1>  1 differences found  datatype: </NamedDatatype> and </NamedDatatype> |
| EXIT-CODE : 1 | EXIT-CODE : 1 |
| ISSUE: Expected to be comparable because the both obj references points exactly same objects. Only different values.  NOTE: -c didn’t do anything | |
|  | ISSUE: Didn’t display difference for /Group/Dset2. However , with ‘-v’, it’s displayed. Refer to verbose output in section ‘2. Optional behaviors’  ISSUE: Didn’t display ”1 differences found” for NamedDatatype. since it’s different is should show as default. (verified from code) |

|  |  |  |
| --- | --- | --- |
| Dataset:  *Region reference* | **$ h5diff ref-dsetreg1.h5 ref-dsetreg2.h5 /Dset\_REGREF**  <NONE> | **$ h5diff ref-dsetreg1.h5 ref-dsetreg2.h5**  dataset: </dset> and </dset>  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. | |

## Continue (default behaviors)

|  |  |  |
| --- | --- | --- |
|  | **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| ***TODO***   * Dataset : obj ref   + Dangle * Dataset : region Ref   + Dangle * Dataset: compound * Dataset: enum (valid, invliad) * Dataset: opaque * Dataset: Vlen * Named Datatype * Group * Attribute: number * Attribute: string * Attribute: obj ref * Attribute: region ref * Soft link * External link * Dangle link | TODO for each | TODO for each |

## Optional behaviors

|  |  |  |
| --- | --- | --- |
|  | **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| -v / –verbose  Dataset : *Number* | $ **h5diff -v dset-num1\_1.h5 dset-num1\_2.h5 /dset /dset**  dataset: </dset> and </dset>  size: [2] [2]  position dset dset difference  ------------------------------------------------------------  [ 0 ] 1 0 1  [ 1 ] 2 0 2  2 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: [2] [2]  position dset dset difference  ------------------------------------------------------------  [ 0 ] 1 0 1  [ 1 ] 2 0 2  2 differences found |
| EXIT-CODE : 1 | EXIT-CODE : 1 |

|  |  |  |
| --- | --- | --- |
| -v / –verbose  Dataset :  *String* | $ **h5diff -v dset-str1\_1.h5 dset-str1\_2.h5 /string /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” | |

### Continue (optional behaviors)

|  |  |  |
| --- | --- | --- |
| -v / –verbose  Dataset :  *space-array* | **$ 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 |

|  |  |  |
| --- | --- | --- |
| -v / –verbose  Dataset :  *type-array* | **$ 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): . | |

## Continue (optional behaviors)

|  |  |  |
| --- | --- | --- |
| -v / –verbose  Dataset :  *Obj Reference* | **$ 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: should be comparable because the both obj references points exactly same objects. Only different values. Also same issue occurs even the first file is copied to be compared. | |
|  | NOTE: -c didn’t do anything | |

## Continue (optional behaviors)

|  |  |  |
| --- | --- | --- |
| -v / –verbose  Dataset :  *Region Reference* | **$ 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. | |

## Continue (optional behaviors)

|  |  |  |
| --- | --- | --- |
|  | **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| ***TODO***   * -v or –verbose : others * -v1 or –verbose=1 * -v2 or –verbose=2 * -r or –report * -q * --follow-symlinks * --no-dangling-links * -N or –nan * -n count or –count= * -d D or –delta D * -p R or –relative=R * --use-system-epsilon * --exclude-path “path” * -- missing RM --- * -c or –compare | TODO for each | TODO for each |

## Handle common, extra or not-comparable object and attribute with results

### Description:

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

1. When difference found in Common object or attribute
2. When difference found because Extra object or attribute exist
3. When difference found because Non comparable object or attribute exist

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

Words definitions:

* **Common object**: object name under root group in two files is same for comparing the two HDF5 files, or object name under two specified group is same for comparing the two HDF5 objects.
* **Common attribute**: attribute name is same between two objects.
* **Extra object**: object name which doesn’t exist under root group in other file for comparing the two HDF5 files; or object name which doesn’t exist under other specified group for comparing the two HDF5 objects.
* **Extra attribute**: attribute name which doesn’t exist under other object.
* **Non-comparable object**: any of type or space is different for comparing two HDF5 object.
* **Non-comparable attribute**: any of type or space is different for comparing two HDF5 object’s attribute.

### 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 |

|  |  |  |
| --- | --- | --- |
|  | **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: 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 | | |

## Insensible behaviors

### Description

This section is for known issues didn’t get covered by the other sections. This section may include same issues from other sections in different point of view.

|  |  |  |
| --- | --- | --- |
|  | **Compare *object* vs *object*** | **Compare *file* vs *file*** |
| Extra object | N/A | $ **h5diff dset-extra1.h5 dset-extra2.h5**  <None> |
| EXIT-CODE : 1 |
| Extra attribute | $ **h5diff extra-attr1.h5 extra-attr2.h5 /do /do**  <None> | $ **h5diff extra-attr1.h5 extra-attr2.h5**  <None> |
| EXIT-CODE : 0 | EXIT-CODE : 0 |

|  |  |  |
| --- | --- | --- |
| Not-comparable  object | $ **h5diff obj-nocomparable1.h5 obj-nocomparable2.h5 /obj1**  --------------------------------  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. |
| EXIT-CODE : 0 | EXIT-CODE : 0 |
| Not-comparable  attribute |  |  |

|  |  |  |
| --- | --- | --- |
| ***TODO***:  Empty dataset  Empty file  Display help page on error  … |  |  |

# Appendix A – List of example HDF5 files

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

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

|  |  |
| --- | --- |
| **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”. |

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

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

# 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 |