RFC: HDF5 Command-line Tool Interface for Handling Attribute Names Containing the Slash Character

Jonathan Kim

The slash ( / ) character is used in HDF5 as an object delimiter, so it is not allowed in object names. Its use is, however, allowed in attribute names.

This can be a problem for the h5dump command-line tool when a user specifies an attribute name containing a slash along with the object path in a single argument.

This RFC fully describes the problem and offers potential solutions.

# Background

The following bug has been reported against h5dump:

h5dump fails to open an attribute when using the -a option to specify an attribute with a slash in the name. (JIRA issue [HDFFV-7523](http://jira.hdfgroup.uiuc.edu/browse/HDFFV-7523))

More specifically, the ‘h5dump -a‘ option allows a user to specify an attribute name in order to display information for only that attribute. The option requires that the attribute name include the full path to the object, a slash (/), and the attribute name. The slash character serves not only as the delimiter between the object path and the attribute name, but also as the delimiter between object names in the object path.

For example:

* h5dump **-a** “/g1/g2/dset**/attr1**” hdf5file.h5

This option works as expected except when the attribute name includes a slash. In such a case, h5dump misinterprets the attribute name and treats the last slash in the attribute name as the delimiter between the object path and the attribute name.

For example:

* Assume that a dataset “/d1” has an attribute “a1/a2”

$ h5dump -a "/d1/**a1/a2**" testfile.h5

HDF5 "testfile.h5" {

ATTRIBUTE "/d1/a1/a2" {

h5dump error: unable to open object "/d1/a1/"

}

}

As can be seen above, h5dump mistakenly determines that the attribute name is the portion of the string that follows the final slash. The rest of the string, that is, everything preceding the final slash, is used as the object path. Since this will not normally identify an existing object, h5dump will normally fail as above. Note that even if such an object is found, h5dump will look for an attribute named by only the last part of the original attribute name; this is also unlikely to exist.

The underlying issue is that h5dump was implemented with an assumption that an attribute name would not contain ‘/’ character, but it had not been fully enforced from the HDF5 lib and long time has passed.

# Suggestions for a general tools interface

This issue must be considered working toward a consistent user interface across the HDF5 command-line tools.

The solutions proposed in this section are for use across multiple tools, not only h5dump.

To accommodate one or more slashes in an attribute name, the object path and attribute name must be specified separately. We see two approaches: two options or one option with two parameters.

***Note***: The short option may change to other character or be removed for only long option. For now “*-O”* and *“-a”* options are used to demonstrate the interface.

## Two options

## **-O “**<object\_path>**” -a “**<attr\_name>**”**

A tool would access an attribute attached to an object with a *-O* option followed immediately by a *–a* option. A tool could access an object with the *–O “<object\_path>”* option only.

**Examples**:

* h5dump -O “/g1/g2/dset1” -a “speed/hour/high” hdf5file.h5
* h5ls -O “/g1/g2/dset1” --attribute=“speed/hour/high” hdf5file.h5

**Long option names**:

* -O or --object-path
* -a or --attribute ( --attribute is used for h5dump and h5stat at this point )

## One option with two parameters

## **-O “**<object\_path>**” “**<attr\_name>**”**

A tool would access an attribute attached to an object with a *-O* option and two separate parameters, the object path and the attribute name. A tool could access an object with *-O “<object\_path>”*, listing only the object path parameter.

This method has been previously discussed with relation to h5diff’s excluding attribute option.

The h5diff’s RFC can be referred from: <https://www.hdfgroup.uiuc.edu/RFC/HDF5/tools/h5diff/ignore_obj/>

**Examples**:

* h5dump –O “/g1/g2/dset1” “speed/hour/high” hdf5file.h5
* h5ls –O “/g1/g2/dset1” “speed/hour/high” hdf5file.h5

# Suggestions for h5dump-only solutions

The following solutions affect no other tools and minimize the disruption to the current h5dump interface.

## Separate object path and attribute name options

h5dump currently uses three options to display objects:

* -d Displays a dataset.
* -g Displays a group.
* -t Displays a committed datatype.

And a single option to display a single attribute:

* -a Displays information for a single attribute.

As discussed above, -a requires a parameter specifying the full path to an object, a slash, and the attribute name.

The suggestion is to allow -a to specify only the attribute name when used with one of the object options. When used alone, -a will retain its current behavior.

**Examples**:

* h5dump -d “/g1/g2/dset1” -a “speed/hour/high” hdf5file.h5

Displays only the attribute “speed/hour/high” of the dataset “dset1”.

* h5dump -g “/g1/g2” -a “speed/hour/mid” hdf5file.h5

Displays only the attribute “speed/hour/mid” of the group “g2”.

* h5dump -a “/g1/g2/attr1” hdf5file.h5

Displays only the attribute “attr1” of the group “/g1/g2”.

## Modify -a to take either one or two parameters: <object\_path> and <attr\_name> separately or the combined <object\_path/attr\_name>

Examples:

Separately:

* -a <object\_path> <attr\_name>
* h5dump -a “/g1/g2/dset1” “speed/hour/high” hdf5file.h5

Combined:

* -a <object\_path/attr\_name>
* h5dump -a “/g1/g2/dset1/seed\_per\_hour” hdf5file.h5

Both of the above would be legal syntax. A user who knows that h5dump might encounter slashes in attribute names might always use the first syntax. A user who knows that the project’s files never use a slash in an attribute name might prefer to use the second syntax. A programmer writing a script for general use could use either, but might be advised to use the first to reduce risk.

## Modify the h5dump parser to search for valid object/attribute combinations

Modify the h5dump parser so that it checks the “object path and attribute name” string for valid object path and attribute name combinations. This modified parser would check the given path/name string by dividing the string at each slash (one at a time) and parsing the resulting combination.

Example situation to consider different outputs

Assume an HDF5 file contains the following objects and attributes:

* The root group (“/”) with an attribute “a/b/c”
* A group “/a” with an attribute “b/c”
* A group “/a/b” with an attribute “c”

In this case, a single -a parameter “/a/b/c” could refer to any of the following object and attribute combinations.

* The attribute “a/b/c” attached to the root group
* The attribute “b/c” attached to the group “/a”
* The attribute “c” attached to the group “/a/b”

### The h5dump could list all valid object path and attribute name combinations that can be derived from the string and that exist in the file.

* As a result, the h5dump would display information for all the above 3 attributes from this example.

### The h5dump’s parser could start at the final slash in the string and work backwards looking only for the first valid object path and attribute name combination.

* As a result, the h5dump would display information for the attribute “c” attached to the group “/a/b” from this example.

## Combination of two preceding approaches

Modify the -a option to take either one or two parameters (per section 3.2) and, when using one parameter, to seek out the first valid object and attribute combination (per the approach in section 3.3).

With this solution, an “h5dump -a <combined\_obj\_path\_&\_attr\_name>” call would have an improved chance of returning a valid outcome. “h5dump -a <object\_path> <attr\_name>” could be used in any situation where that returned value is not the desired outcome or greater specificity is desired.

# Recommendation

How to specify attribute for a command-line tool had been a topic to discuss in the past and it’s time to discuss more in considering general tools. We also should consider the direction of the choice could affect user’s impression differently about the relationship between object and attribute. For example specifying attribute separate from object could guides user’s impression that an attribute is different from an object, which could help to reflect those definitions more clearly.

The same topic for the h5edit may needs to be discussed independently since it’s interface style is totally different from other command-line tools.

# Revision History

|  |  |
| --- | --- |
| *September 21, 2011:* | Version 1 circulated for initial feedback and update ( Peter & Frank ) |
| *September 28, 2011:* | Version 2 circulated for internal discussion and feedback |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |