

## Review of Mapping HDF5 to DAP2 - Technical Note

The goals of the Standards Process Group (SPG) of NASA's Earth Science Data Systems Working Groups are to:

1. Enable data and service providers to easily join NASA's Earth Science network of data systems through use of standards.
2. Facilitate interoperability between components of NASA's Earth Science network of data systems through use of standards.
3. Facilitate data stewardship and preservation through use of standards and adoption of best practices.
4. Develop and manage effective standards recommendation, adoption, and approval processes to guide the evolution of ESDS standards. Support the evolving strategies and goals of NASA's Earth Science activities through use of standards.

One of the ways we do this is by publishing Technical Notes relevant to Earth Science Data Systems. An SPG Technical Note is a document that contains important and useful information that is relevant to the domain of NASA Earth Science Data Systems, and does not necessarily describe a "standard" which has additional operational requirements. In order to assure a high level of technical quality, we conduct public reviews of Technical Notes that have been submitted to us for consideration.

We are asking you to review the Technical Note referenced below. Your assistance will help us to decide whether each it should be endorsed by the SPG.

- ESDS-RFC-017 - Mapping HDF5 to DAP2

**<http://www.esdswg.net/spg/rfc/esds-rfc-017/ESDS-RFC-017v0.1.pdf>**

You are invited to review this technical note and provide feedback that might make the document more useful (see review questions below).

1. Please provide your name, organization and contact information including e-mail address. (*This information will not be shared.*)

2. Are you answering for your entire organization, for a smaller group, or individually?

\_\_\_\_\_ a) Entire organization

\_\_\_\_\_ x \_\_\_\_\_ b) Smaller group (please specify) \_\_\_\_\_

\_\_\_\_\_ c) Individual response

3. Describe in a sentence or two your overall experience related to HDF5 or DAP:

*(e.g., specification developer, specification implementer, systems architecture; tools developer, scientific analysis; etc.)*

*A is a maintainer of the HDF libraries with The HDF Group and A also developed HDF5 Fortran library. B is a ... of The HDF Group. B has been peripherally involved with the HDF5-OPeNDAP project, and participate in early discussions about the mapping with Unidata and OPeNDAP.*

4. Do any of your systems currently use (or expect to use) HDF5 or DAP?

*Our group develops and maintains HDF4 and HDF5 software. We also work with OPeNDAP on integrating HDF4 and HDF5 OPeNDAP and deploying the resulting products.*

5. Does the technical note contain internal inconsistencies? If so, please provide details.

*Yes. Several examples are given below. For more detailed discussion see the Appendix to this document.*

*The Abstract, as well as the “Motivation and purpose,” states that the technical note presents “**general** HDF5-to-DAP2 mapping information.” This may be misleading, as it may be construed to mean that the document shows how all HDF5 objects can or cannot be mapped to DAP2. In fact, the paper’s main focus is a mapping of a **subset** of the HDF5 objects used in the HDF-EOS5 and NPOESS data models to DAP2.*

*The paper also appears to claim in one case (section 3.2.1) that the HDF5 objects outside the discussed subset cannot be mapped to DAP2, which we believe is not the case (see comments in Appendix).*

*The document is based on DAP2 and HDF5 standard documents, but the authors are inconsistent in using the terminology and notations from those documents; for example, “data type” on page 6 vs. HDF5 standard’s “datatype” when referring to the object, and “Uin16” vs. DAP2’s “UInt16”.*

*The authors give their own definitions of “dataset” in introductions to HDF5 and DAP2 objects on page 6, and do not use the definitions from the HDF5 and DAP2 standard documents. The second sentence in section 2.2 presents a DAP2 dataset as an object that consists of data types, whereas the DAP2 standard describes a DAP2 dataset as “a container that encompasses all the variables provided in some data source” (see page 10, ESE-RFC-004v1.1.pdf).*

6. Are any parts of any of the technical note ambiguous or poorly explained? If so, please provide details.  
*Yes. Several examples are given below. For more detailed discussion see the Appendix to this document.*

***Choice of mappings and methods used.*** *In section 3, we feel the paper needs to do a better job of explaining the context of how the mappings were arrived at and what this particular presentation means. That is:*

*The authors use examples to explain the HDF5-to-DAP2 mappings, but do not provide descriptions of how these mappings are created. For example, the mapping of an entire hierarchy of groups and datasets in an HDF5 file to DAP2 is illustrated on pages 14 and 15, but it is not clear how that mapping was generated. A general description of how mappings are generated would make it possible to see, for instance, how the DAP2 DAS table on page 15 would change if the HDF5 file on page 14 had a different structure, e.g., if group g2 has several child groups with the groups and datasets underneath.*

*Indeed, in a few cases we had difficulty understanding why the authors had chosen the particular mappings that they chose. If this were explained, it would help the reader in assessing whether these were the best options, or alternatives might have been selected. It was also unclear whether there was a methodology for creating mappings, or whether the mappings were created in an ad hoc manner. We have no issue with the latter approach, but we feel the authors should clear about this. If no formal methodology exists for generating mappings, it would still be good to indicate how these mappings were arrived at. This might help address our issue with section 3.2.1, in which the paper claims that certain datatypes cannot be mapped.*

*As they are presented, some examples seem to be presented as rules for creating mappings. This may not have been the intention, and if it wasn't, then the paper should state that this is a selection of examples of how mappings can be done, and not a set of rules for creating mappings.*

**Terminology.** In section 3.2.2.3 Compound data type, a DAP2 DDS is shown, but the corresponding HDF5 compound datatype is not. The HDF5 dataset is also referred to as “HDF5 compound data type array”, which is not the normal way to refer to an HDF5 dataset. (If the authors use “array” and “dataset” interchangeably, this should be stated at the beginning of the document to avoid confusion, especially since these terms are used in both standards and have different meanings.)

**Discussions of limitations.** The authors do not discuss limitations that may affect the mappings. For example, the number of elements in DAP2 Array to which HDF5 dataset is mapped to, **MUST NOT** exceed  $2^{31} - 1$  (see page 1010, ESE-RFC-004v1.1.pdf), but the number of elements in HDF5 dataset can routinely exceed this limit. Thus, mapping an HDF5 file structure to DAP2 attribute (see page 15) will work for most HDF-EOS5 and NPOESS files, but may not work for a general HDF5 file with a large number (thousands, millions) of groups and datasets.

7. Did you find the technical note useful and would you like to see more such technical notes processed by the SPG?

*Yes. We found the technical note to be very useful since it shows possible mappings from HDF-EOS5 and NPOESS to DAP2. The mapping was implemented as a part of feasibility study and is invaluable for any future HDF5-DAP2 work.*

8. Should the SPG endorse this document as a Technical Note – why or why not?

*Yes, if the document is reworked and its intention is made clearer. We are especially concerned that, as currently written, readers may assume that these mappings are ready to be used in real applications.*

**Note:** The SPG has already endorsed DAP2 and HDF 5 as standards. This question pertains strictly to whether this proposed Technical Note should be endorsed.

## **Appendix:**

### **Page 1 (editorial)**

Fonts are not consistent between the headings and the text (font size and style). Later in the document the same font and size is used in the headings, but headings should probably be in **bold** to make the document consistent with the ESE template standard.

As was mentioned in section 5, the Abstract doesn't clearly (or correctly) explain the intention of the technical note.

The authors use quotes around "group" in "HDF5 'group' object", which seems to be unnecessary and doesn't match the usage of the term in the HDF5 standard document ES<sub>DS</sub>-RFC-007v1.

In "Status of this Memo" "Mapping" should probably be "mapping".

### **Pages 3-4 (editorial)**

Two different spellings: "non-profit" on page 3 and "nonprofit" on page 4.

### **Page 6**

An HDF5 link is NOT really like a "path" as it is described in the second paragraph. I would suggest that authors use "local path" or use explanation from the HDF5 standard document ES<sub>DS</sub>-RFC-007v1.

The authors' definition of compound datatype as a contiguous sequence of bytes in the third paragraph is hard to understand; for example, how then it is different from a 4-byte integer or array of floats? The key here should be a similarity with a heterogeneous record structure, such as a C structure, and not a sequence of bytes.

The inconsistent usage of term "data type" and ambiguity of term "dataset" has already been mentioned.

In section 2.2 DAP2 names for the 16-bit unsigned short integer should be U<sub>Int</sub>16 and for the 32-bit unsigned integer U<sub>Int</sub>32 according to the DAP2 standard ESE-RFC-004.1.1. The same correction should be done in Table 1 on page 8.

### **Page 7**

In section 3 the authors write that some HDF5 objects can be difficult or even impossible to map to DAP2, but they do not discuss why and what is difficult. In contrast, the paper demonstrates a very successful approach of mapping the complex HDF-EOS5 objects such as swaths and grids that are built using HDF5 objects to DAP2. The same approach can be used to map the "difficult" HDF5 objects. The authors should say why they didn't use that approach here. Simply claiming that some objects cannot be mapped is misleading.

### **Page 8**

Table 1 shows the mapping between HDF5 datatypes and DAP2 data types. DAP2 data types are XDR-based (IEEE big-endian), but HDF5 datatypes are not restricted to the XDR types. Also the HDF5 standard document ES<sub>DS</sub>-RFC-007v1 doesn't use the notation for HDF5 datatypes shown in the table. Mapping between DAP2 XDR types and HDF5 types that are not IEEE and/or big-endian can be explained better.

As was mentioned above, there are typos in the names of the DAP datatypes in Table 1 (U<sub>Int</sub>16 and U<sub>Int</sub>32). Table 1 should probably use "DAP2 Data type" as it is in Table 2 on page 30.

In section 3.2.1 the authors discussed HDF5 atomic datatypes that cannot be mapped to DAP2. Maybe the intention was to list the datatypes that they didn't consider in their work. Or perhaps there are other constraints on what constitutes a valid mapping. Simply claiming that those datatypes cannot be mapped is misleading. One can come up with mappings such as the following (although these may not be the best):

- The HDF5 bitfield datatype can be mapped to one of the DAP2 integer types; an attribute can be created to pass additional information about HDF5 type (endianess, size, etc.).
- The HDF5 opaque datatype can be mapped to the DAP2 Byte; an attribute can be created to pass additional information about the type.
- HDF5 doesn't officially support time type (but it is an integer type, so it can be easily mapped if required).
- The HDF5 enumeration type is based on the integer type. It can be mapped to one of the DAP2's integer types; an attribute can be created to pass <name, value> pairs.
- An HDF5 dataset that has N dimensions with an array datatype (M dimensions) can be mapped to DAP array variable of M+N dimensions; additional information about datatype can be passed via a DAP2 attribute.
- Elements of HDF5 variable length datatype can also be presented by elements of DAP2 atomic and constructor data types in many different ways, for example, sequence of structures, where structure contains the length and variable-type data itself.

## Pages 9 – 11

The authors have chosen DAP2 URL type to present HDF5 object and region references and it is an absolutely valid approach. But several things should be considered before recommending this particular mapping:

1. According to DAP2 standard ESE-RFC-004.1.1 page 9 section 3.2.3, the URL type has a specific meaning of a pointer to some WWW resources; the document also says that it can be used to refer to another data source. The authors treat an HDF5 file as one data source, and since the referenced HDF5 objects have to be in the same file as the references pointing to them, the usage of a URL seems to be inconsistent with the DAP2 standard.
2. HDF5 version 1.8.0 introduced external links. By using an external link, an HDF5 group, for example, may contain an object that "resides" in another HDF5 file. Design of the HDF5 external links doesn't limit them to point to HDF5 objects. One can implement an external link as a URL and "teach" HDF5 library how to access the data referred by URL. Using DAP2 URL for the object and region references may make it difficult to map external links in the future.
3. One cannot distinguish between an HDF5 dataset with object references from an HDF5 dataset with region references and has to retrieve data to find the difference (see examples of DDS on pages 10 and 11).
4. HDF5 allows UTF-8 encoding for links (datasets and groups names) and attributes names. A URL is restricted to the ASCII character set only. This is one of the limitations that should be discussed for the mapping in general and not only for URL.

## Page 12

HDF5 region references can point to a union of several sub-arrays. It looks like the authors addressed only region references that point to just one sub-array. The syntax of a region shown in 4) on page 12 needs to be reworked to address the general case, or at least this shortcoming should be mentioned.

Mapping of the HDF5 datasets is described briefly in the section 3.3. Unfortunately the authors doesn't discuss datasets with scalar and null dataspace (e.g., datasets with just one value that IS NOT a vector of size 1 and datasets that are not intended to have data at all (common usage is in netCDF-4 as a placeholder for an attribute.)) Without availability of the spatial information for datasets /g3/d1 and /g3/d4, the DDS on page 15 becomes ambiguous.

### **Page 13**

As mentioned above, the proposed approach to mapping HDF5 groups may not be scalable.

### **Page 17**

It is not clear why the attribute mapped from a group attribute must be distinguished from the attribute mapped from a dataset attribute, and therefore an extra "/" is needed in the DAS table. Since the HDF5\_ROOT\_GROUP attribute contains the structure of the HDF5 file where g4 will be shown as a group, and since there cannot be a dataset with the same name in the parent(s) of the group g4, one can easily distinguish the attributes between datasets and groups. But this distinction may be not necessary at all.

A more formal approach would be more than helpful in illustrating the mapping. It is possible that the authors are missing something in their approach, and a more formal approach would help determine this.

It would be also helpful if the example contained more than one attribute to a group or a dataset to illustrate the DAS table.

### **Pages 19 – 29**

The authors did a very good job explaining mappings of the HDF-EOS5 objects to DAP2. Using DAP2 attributes to pass objects' metadata and standard DAP2 atomic and constructor variables to pass data seems like a very good approach and can be used to map most HDF5 objects to DAP2.

The paper will benefit greatly by using some kind of formal notation (or graphical presentation) to describe the mappings. It is not easy to generalize most of the examples provided.

### **Page 30**

DAP2 types UInt16 and UInt32 should be fixed. Different entries in the table use different fonts.