HDF5-DODS Data Model and Mapping

Author: MuQun Yang and Robert E. McGrath
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
July 5, 2001

Contents

1. Introduction

This document describes how to represent an HDF5 file according to the Distributed Oceanographic Data System (DODS) model. This is the conceptual design from which the DODS server can be constructed. This mapping must provide a definition of standard mappings from the objects in an HDF5 file to DODS objects in DAS and DDS records. Please refer to the HDF5 ([3]) and DODS ([6]) documentation.

The Distributed Oceanographic Data System (DODS) [1] uses ancillary data to describe the shape and size of data types that make up the data set, and provides information about the dataset's attributes, as well. The Dataset Descriptor Structure (DDS) describes the data set's structure and the relationships between its variables, and the Dataset Attribute Structure (DAS) provides information about the variables themselves. DODS server must generate DDS and DAS when doing data transfer. The DODS client can then retrieve the data based on DDS and DAS [6].

In order to use HDF5 with the DODS, a data server must be written using the DODS toolkit [6]. This software is compiled with DODS support code, and is run by the DODS server. The HDF5-

DODS server is responsible for reading HDF5 files and constructing appropriate DODS data structures by calling DODS library routines.

The HDF5 model [3] is more general than DODS [6], and the overlapping concepts cannot be perfectly matched. While most HDF5 concepts have a corresponding concept in DODS, some might be represented in more than one way in DODS, and some have no corresponding DODS representation at all. Table 1 gives examples of HDF5 concepts in three categories.

Table 1 shows three categories of HDF5 concepts, cases A, B, and C. Case A are concepts that are clearly analogous, such as HDF5 datasets with numeric or string data. These map to DODS arrays. Case B are HDF5 concepts for which there may be one or more corresponding DODS objects, but it is not completely clear if there is a single correct mapping. For example, HDF5 compound data types appear to be analogous to DODS Structures, but it is not certain that this is always appropriate. Also, a one-dimensional array of compound data might well be mapped instead to a DODS Sequence.

Case C are concepts from HDF5 that have no corresponding concepts in the DODS model. These include HDF5 Groups and Named Datatypes. In most cases, these could be represented in DODS through conventions, such as special attributes. However, "normal" DODS clients would not necessarily know these conventions, and the information might be difficult for non-HDF5 applications to fathom.

Table 1. Conceptual mapping from HDF5 to DODS

Case	**Examples of HDF5 concepts**
A. A clear mapping exists	Datasets with atomic number type, Datasets with String type. Attributes of datasets with scalar types
B. Ambiguous, multiple, or uncertain mappings	One-dimensional datasets with non-nested compound data types. General compound data types, array types.
C. No correspondence	Groups, Named data types. Other non-atomic types, e.g., object references.

In the mapping that follows, all of the correspondences of Case A are specified. Concepts in Case B and Case C are mostly not specified. The mapping has two main parts. Section 2 defines a mapping of HDF5 Objects (datasets, etc.), and Section 3 maps HDF5 data types to DODS data types. Section 4 discusses important limitations in this mapping.

2. Object Mapping

This section specifies the recommended mapping for the main HDF5 objects, the dataset and the group.

2.1. HDF5 Datasets and Attributes

As a general rule, an HDF5 dataset will map to a DODS ARRAY, with a corresponding DODS data type. The attributes of the HDF5 dataset are used as DODS attributes. If the HDF5 dataset

has dimension scales (e.g., a dataset converted from an HDF4 SDS using the *h4toh5* utility [7]), should be mapped to a DODS GRID. Table 2 lists these mappings.

Table 2. Mapping of HDF5 object to DODS objects.

HDF5 Object	**DODS Object**
Dataset, type is Integer or Float	ARRAY of appropriate DODS number type
Dataset, type is String	ARRAY of DODS string
Datasets with appropriate dimension scales, as implemented in files converted from HDF4 SDS	DODS GRID, with appropriate DODS number type
Dataset Attribute	Attribute on equivalent DODS object
Dataset, type is Compound	ARRAY of DODS structure (?) If 1D, then DODS sequence (?)

Table 2 lists two possible mappings for datasets with compound data type. It appears that some compound datasets conceptually map to a DODS ARRAY of Structures, and some may best be considered a DODS Sequence.

2.2. HDF5 Groups

An HDF5 file is a rooted directed-graph, with at least one group, defining an appropriate data mapping from HDF5 file structure to DODS is not trivial. DODS does not have a concept that corresponds to HDF5's grouping structure. DODS generally treats a file as a flat set or sequence of datasets, with no grouping or nesting structure. Furthermore, many DODS applications don't care or simply cannot understand a structured file. Information about the file structure is likely to be irrelevant to these applications.

We considered two options for representing HDF5 groups with auxiliary DODS records.

2.2.1. Option #1: A DODS structure

The DODS structure can be used to represent the grouping. The HDF5 group would be treated as a single object, all the objects within it treated as elements of the structure. Attributes of groups could be represented naturally as attributes of this structure. This is how the HDF4-DODS server handled HDF4 Vgroups.

In this option, the structure is a tree. Any loops or multiple links to the same object can't be directly represented.

The disadvantage of this approach is that every HDF5 file will contain exactly one object, the structure representing the root group. This structure will contain all the HDF5 objects within it. This is not very intuitive, and DODS applications may be confused. Most DODS applications expect a list of datasets, not one big, complicated structure.

2.2.2. Option #2: A DODS attribute

The grouping structure of the HDF5 file could be ignored when mapping to DODS. All the HDF5 datasets will be used with no groups at all. This will be natural for most DODS applications, the file will be presented as a set of datasets.

In this option, information about the group structure can be included as DODS attributes. For example, we can choose an unambiguous absolute name path for individual dataset, and store this as an attribute. The other features of the group structure are ignored, so obviously this mapping is not complete.

There are two variations for how the grouping information might be included. One would be to include a single DODS structure, representing the groups of the HDF5 file in some convention. DODS applications could ignore this variable if they don't need it, and could decode it to determine the group structure of the HDF5 file.

An alternative would be to encode a description of the group structure in a DODS attribute. Again, applications could ignore this attribute or decode it as needed.

In these variations, the DODS data contains one object for each HDF5 dataset, plus either an extra object or an extra attribute to represent the group structure.

2.2.3. Summary

It is not clear which approach would be best. Of paramount importance is the question of what DODS client applications expect, and how these options would affect them.

3. HDF5 Data Type Mappings

3.1. Atomic Number Types: Integer and Floating Point

HDF5 Atomic data types of class Integer and Float will be mapped to equivalent DODS numeric data types. Table 3 gives the corresponding types for HDF5 atomic numeric types. Some types have no equivalent in DODS. In some cases, it may be acceptable to applications to use an inexact mapping (e.g., Signed 8-bit to Byte), or even to convert data, e.g., from 64-bit Integers to 64-bit Floats). These decisions cannot be made without knowledge of the specific datasets.

3.2. HDF5 String

HDF5 String will be converted into DODS string.

3.3. Other Atomic Types with no Mapping to DODS Types

The following HDF5 data types have no corresponding type DODS types. Datasets and attributes with these types cannot be represented in DODS.

TIME
BITFIELD
OPAQUE
ENUM
REFERENCE

Table 3. Mapping of HDF5 Atomic Number Types to Equivalent DODS Data Types

HDF5 Data Type	DODS Data Type
Signed 8 bit Integer	<no equivalent type>
Unsigned 8 bit Integer	Byte
Signed 16-bit Integer	Int16
Unsigned 16-bit Integer	Uint16
Signed 32-bit Integer	Int32
Unsigned 32-bit Integer	Uint32
Signed 64-bit Integer	<no equivalent type>
Unsigned 64-bit Integer	<no equivalent type>
32-bit Float	Float32
64-bit Float	Float64

3.4. HDF5 Compound Data

HDF5 compound data types are a very general class of structured records. The DODS data model includes two kinds of structured records, DODS Structure and DODS Sequence. For one dimensional data, both Structure and Sequence correspond to a one-dimensional HDF5 dataset with compound data types (which may be nested); which the atomic elements are types that as mapped above. It is uncertain when a Structure would be preferred over a Sequence. For multidimensional arrays of compound data, a DODS ARRAY of Structures is the only corresponding data type.

3.5. Other HDF5 Datatypes with No Clear Mapping

The following HDF5 Datatypes have no mapping in DODS.

ARRAY
VLEN

In some cases the HDF5 Dataset with these type might reasonably be mapped to a DODS ARRAY. For example, an HDF5 Dataset with elements that are one-dimensional ARRAYs of atomic number types might reasonably be mapped to a DODS ARRAY with one more dimension than the HDF5 Dataspace. Similarly, an HDF5 Dataset of VLEN of type String, could be mapped to a DODS ARRAY of Strings, with appropriate padding if needed.

3.5. Summary

The HDF5 Data model includes many Datatypes not supported by DODS. Fortunately, multidimensional arrays of numbers and Strings can be mapped easily, which are by far the most important. Other types may have mappings that might be used for some datasets.

4. Limitations and Warnings

Because of the essential differences between HDF5 and DODS, Mapping from HDF5 to DODS is impractical or even impossible for some objects and some data types. Some of these limitations were discussed in Section 2 and 3. This section explains other important limitations of the HDF5 to DODS mapping.

4.1. Special Characters HDF5 Names

A number of non-alphanumeric characters (e.g. space, #,+,-,/) allowed in HDF5 path names are not allowed in the names of DODS objects or object components nor in URLs. These characters should be escaped using the Web CGI conventions [4].

The special character like "/" is especially important because HDF5 path names use "/" as a delimiter, but this cannot be use in a DODS attribute. This means that the unambiguous full name of the HDF5 object can't be used as a name attribute.

One approach is to use the relative object name (which may not be unique in the HDF5 file) as the name attribute in DAS and DDS and to create an additional DODS attribute called HDF5_OBJ_FULLPATH to store the full path name as a string.

4.2. HDF5 Attributes

DODS attributes as wither a scalar or a one-dimensional array. The DODS DAS does not even give information to show the size of the attribute if the attribute is a one-dimensional array. Also, all DODS attribute data must be converted into DODS string no matter what the original attribute data type.

HDF5 attribute data can be a multi-dimensional array of any data type. Any attribute that cannot be represented in DODS will be omitted.

4.3. Multiple Links to HDF5 objects

There is no equivalent DODS data structure to represent HDF5 group with loops, and there are no simple ways to represent HDF5 soft and hard links in DODS. It would be necessary to invent a convention for representing these cases in a DODS attribute or structure.

5. Summary

This document specifies a mapping between the concepts of HDF5 and DODS. Some concepts can be mapped to clearly corresponding DODS concepts. Others have no mapping, or an unclear mapping. Most DODS applications "expect" data that falls in the cases that can be clearly mapped. The specification in this document can be used to create a DODS HDF5 server.

Acknowledgements

This project was funded by NASA Cooperative Agreement, NASA NCC5-307. Any opinions, findings, and conclusions or recomendations expressed in this material are those of the author(s), and do not necessarily reflect the views of the National Aeronautics and Space Administration.

References

1. “DODS”, http://www.unidata.ucar.edu/packages/dods/

2. “HDF5 - A New Generation of HDF”, http://hdf.ncsa.uiuc.edu/HDF5

3. “HDF5 Abstract Data Model”, http://hdf.ncsa.uiuc.edu/HDF5/papers/ADM/ADM_EOS_Sep99/EOSpresentation/index.html

4. T. Berners-Lee, R. Fielding, L. Masinter, “Uniform Resource Identifiers (URI): Generic Syntax”, IETF RFC 2396, August 1998.

5. Tom Sgouros, “DODS User Guide Version 1.9, March 28, 2000. http://www.unidata.ucar.edu/packages/dods/user/guide-html/

6. James Gallagher, Tom Sgouros, “DODS Programmer's Guide: The Client and Server Toolkit Version 1.2”, March 23, 2000. http://www.unidata.ucar.edu/packages/dods/api/pguide-html/

7. *h4toh5*, http://hdf.ncsa.uiuc.edu/HDF5/doc/Tools.html#Tools-H4toh5