HDF5 Metadata: Library and User

Author: Frank Baker

Version 2

This document presents two discussions of HDF5 metadata:

* A high level view derived from the text that Peter, Quincey, and I developed for Peter's h5diff comparison specification  document. This might suitable for a user guide chapter.
* A more detailed and technical view that is intended for inclusion in "Advanced Topics."

These texts experiment with terminology.  Library metadata has previously been discussed as "structural metatdata," but user metadata can also be structural; the key is that this metadata is automatically generated by the library.  The two categories of user metadata used to be discussed jointly as "application metadata" without differentiating static and dynamic metadata.   Static user metadata is normally set only once and often cannot be changed without rewriting an object or file, while dynamic user metadata is designed to be changed at will.

# HDF5 Metadata -- Overview

HDF5 files generally contain two types of metadata: library metadata and user metadata.

Library metadata is generated by the HDF5 Library to describe the structure of the file and structure and contents of objects in the file.  For example, libray metadata includes information such as:

* A header block (superblock) that sets up the file, sets up the initial structures, and identifies the file as a valid HDF5 file
* B-trees that describe the location of and provide access to groups and members of groups
* Datatype, current and maximum array dimensions, and other features of a dataset
* Dataset properties such as storage layout, fill value, allocation time, or the use of filters

HDF5 natively interprets and understands library metadata.  Library metadata is always present; even an otherwise-empty file must contain certain metadata to be a valid HDF5 file.

User metadata is defined and provided by a user application, is often stored in an HDF5 attribute, and may describe virtually anything.  Examples include:

* Minimum and maximum valid values in a dataset
* Conditions under which data was collected
* Data history and/or provenance
* Relationships among datasets
* Scales or other interpretive information

HDF5 does not natively understand all forms of user metadata; such metadata must be understood and interpreted by the application.  User metadata is optional but very commonly used.

# HDF5 Metadata -- Detailed Review

HDF5 files can contain several types of metadata:

* Library metadata
* Static user metadata
* Dynamic user metadata

## Library Metadata

Library metadata is metadata that the user doesn't have any interaction with or control over.  It is generated by the HDF5 Library to describe the structure of the file and structure and contents of objects in the file.  For example, library metadata includes information such as:

* Most elements of the header block (superblock), which sets  up the file, sets up the initial structures, and identifies   the file as a valid HDF5 file
* Many elements of object headers, which set up objects in anHDF5 file, including:
* Datatype, current and maximum array dimensions of a dataset, etc.
* Dataset properties such as storage layout, fill value, allocation time, or the use of filters
* B-trees that describe the location of and provide access to groups and members of groups

HDF5 natively interprets and understands library metadata.  Library metadata is always present; even an otherwise-empty file must contain a superblock and a root group object header to be a valid HDF5 file.

## Dynamic and Static User Metadata

User metadata is defined and provided by the user application.

### Static user metadata

Static user metadata is information that the user has control over but that is not generally dynamic.  It is stored in the file superblock, an object header, a property list, or ???, and it does not usually change through the life of a file or object.  Examples of static user metadata include:

* Property lists: For example, H5Pset\_fapl\_sec2 specifies in the file access property list that file I/O will use the sec2 driver.
* Link names
* A dataset's datatype and dataspace (modulo the potential to extend/shrink it)
* Dataset fill values
* Dataset or group storage properties

Static user metadata is not usually changed through through the life of a file or object.  In some cases, it just doesn't tend to change; the name of a the hard link to an object is  making a new copy of an HDF5 file or object.  In some cases, that is the only time it is possible to change it; , for example, file access and dataset access properties, for example, can be only changed only when making a new copy of a file or dataset, respectively.

### Dynamic user metadata

Dynamic user metadata is metadata that the user or application can change at will.  It is often stored in an HDF5 attribute, may describe virtually anything, and can easily change over time.  For example:

* Minimum and maximum valid values in a dataset
* Conditions under which data was collected
* Data history and/or provenance
* Relationships among datasets
* Scales or other interpretive information

HDF5 does not natively understand application metadata; it must be understood and interpreted by the application.  Application metadata is technically optional but commonly used.