Discovering the Contents of an HDF5 File

Discovering what is in an HDF5 file

HDFView and h5dump are standalone tools which cannot be called within an application, and using H5Dopen and H5Dread require that you know the name of the HDF5 dataset. How would an application that has no prior knowledge of an HDF5 file be able to determine or discover the contents of it, much like HDFView and h5dump?

The answer is that there are ways to discover the contents of an HDF5 file, by using the H5G, H5L and H5O APIs:

- The H5G interface (covered earlier) consists of routines for working with groups. A group is a structure that can be used to organize zero or more HDF5 objects, not unlike a Unix directory.
- The H5L interface consists of link routines. A link is a path between groups. The H5L interface allows objects to be accessed by use of these links.
- The H5O interface consists of routines for working with objects. Datasets, groups, and committed datatypes are all *objects* in HDF5.

Interface routines that simplify the process:

- H5L_ITERATE1 traverses the links in a specified group, in the order of the specified index, using a user-defined callback routine. (A callback function is one that will be called when a certain condition is met, at a certain point in the future.)
- H5O_VISIT / H5L_VISIT1 recursively visit all objects/links accessible from a specified object/group.

Programming Examples

Using H5Literate, H5Lvisit and H5Ovisit:

Under HDF5 Examples you will find the examples **By API**, where examples of using H5Literate and H5Ovisit/H5Lvisit are included.

The *h5ex_g_traverse* example traverses a file using H5Literate: C F90

The *h5ex_g_visit* example traverses a file using H5Ovisit and H5Lvisit: C F90