

HDF5 Lite (H5LT) Interface

With HDF5 there are a number of steps required to create a dataset or attribute. The HDF5 Lite interface condenses these steps into a single call.

Even with the High Level APIs, the more general HDF5 create and close functions may or will still need to be used. In particular, please refer to the introductory tutorial topic on [creating a file](#). You may also want to review the topic on [creating a dataset](#).

Covered here are the different HDF5 Lite calls available for writing to and reading from [Datasets](#) and [Attributes](#).

Datasets

Writing a Dataset

To create and write a dataset, use the general function `H5LTmake_dataset`:

```
H5LTmake_dataset (file_id, dset_name, rank, dims, H5T_NATIVE_INT, data);
```

This function accepts a parameter `file_id`, obtained from the basic HDF5 library function, `H5Fcreate`, a dataset name, the number of dimensions in the dataset, an array containing the dimensions, the HDF5 datatype (`H5T_NATIVE_INT`) for the data, and the data.

PROGRAMMING EXAMPLE

The following example demonstrates how to create and write a dataset using the HDF5 Lite function `H5LTmake_dataset`.

C Example

There are other Lite functions that allow the writing of a dataset. These functions are type specific; there is one of them for each of the most common HDF5 datatypes. These functions are listed below. For example, to write an integer array of data, use the `H5LTmake_dataset_int` function:

```
H5LTmake_dataset_int (file_id, DSET3_NAME, rank, dims, data_int_in);
```

The parameters of this function are similar to `H5LTmake_dataset`, except that it does not include the parameter for the datatype. The common types that have specific make functions are the following predefined native datatypes:

C language type	Function	HDF5 type
string	<code>H5LTmake_dataset_string</code>	<code>H5T_C_S1</code>
char	<code>H5LTmake_dataset_char</code>	<code>H5T_NATIVE_CHAR</code>
short	<code>H5LTmake_dataset_short</code>	<code>H5T_NATIVE_SHORT</code>
int	<code>H5LTmake_dataset_int</code>	<code>H5T_NATIVE_INT</code>
long	<code>H5LTmake_dataset_long</code>	<code>H5T_NATIVE_LONG</code>
float	<code>H5LTmake_dataset_float</code>	<code>H5T_NATIVE_FLOAT</code>
double	<code>H5LTmake_dataset_double</code>	<code>H5T_NATIVE_DOUBLE</code>

Reading a Dataset

To read back the data there are the opposite functions: a generic read function that accepts an HDF5 type parameter,

```
H5LTread_dataset (file_id, dset_name, H5T_NATIVE_INT, data);
```

and the more specific functions for the more common types. The following call achieves the same results as the above call:

```
H5LTread_dataset_int (file_id, dset_name, data);
```

In the case of the read functions, obtain the HDF5 file identifier from the basic library function, `H5Fopen`.

PROGRAMMING EXAMPLE

The following example demonstrates how to read a dataset using the HDF5 Lite function `H5LTread_dataset_int`:

C Example

Similar to the make dataset functions, the common types that have specific read functions are the following:

C language type	Function	HDF5 type
string	<code>H5LTread_dataset_string</code>	<code>H5T_C_S1</code>
char	<code>H5LTread_dataset_char</code>	<code>H5T_NATIVE_CHAR</code>
short	<code>H5LTread_dataset_short</code>	<code>H5T_NATIVE_SHORT</code>
int	<code>H5LTread_dataset_int</code>	<code>H5T_NATIVE_INT</code>
long	<code>H5LTread_dataset_long</code>	<code>H5T_NATIVE_LONG</code>
float	<code>H5LTread_dataset_float</code>	<code>H5T_NATIVE_FLOAT</code>
double	<code>H5LTread_dataset_double</code>	<code>H5T_NATIVE_DOUBLE</code>

Attributes

Writing an Attribute

Similar to the Lite write dataset functions, there are several Lite write attribute functions, one for each HDF5 datatype. For example, to write an integer attribute, the function `H5LTset_attribute_int` is used. The use of this function is:

```
H5LTset_attribute_int (file_id, dset_name, attr_name, data, size);
```

This function accepts a parameter, `file_id`, obtained from the basic HDF5 library function `H5Fcreate` or `H5Fopen`, the object (dataset or group) name in which we want to create the attribute, the data and its array size.

Reading Attributes

To read an attribute, the steps are similar, except they are read functions:

```
H5LTget_attribute_int (file_id, dset_name, attr_name, data);
```

PROGRAMMING EXAMPLE

The following example demonstrates how to write and read an attribute using the HDF5 Lite functions `H5LTset_attribute_int` and `H5LTget_attribute_int`.

C Example

The more common Lite functions that allow the creating of attributes are listed below. These functions are type specific; there is one for each of the most common HDF5 datatypes. There are similar functions for reading.

C language type	Function	HDF5 type
string	<code>H5LTset_attribute_string</code>	<code>H5T_C_S1</code>
char	<code>H5LTset_attribute_char</code>	<code>H5T_NATIVE_CHAR</code>

short	H5LTset_attribute_short	H5T_NATIVE_SHORT
int	H5LTset_attribute_int	H5T_NATIVE_INT
long	H5LTset_attribute_long	H5T_NATIVE_LONG
float	H5LTset_attribute_float	H5T_NATIVE_FLOAT
double	H5LTset_attribute_double	H5T_NATIVE_DOUBLE