H5P_ITERATE

Expand all Collapse all

- Jump to ...
- Summary
- Description
- Example
- Switch language ...
- C
- C++
- FORTRAN
- JAVA

Summary
Description
Example
JAVA
FORTRAN
C++
C

H5P_ITERATE

Iterates over properties in a property class or list

Procedure:

H5P_ITERATE (id, idx, iter_func, iter_data)

Signature:

```
int H5Piterate(
               hid_t id,
               int * idx,
               H5P_iterate_t iter_func,
               void * iter_data
 )
```

```
NONE
```

Parameters:

hid_t `id`	IN: Identifier of property object to iterate over
int * `idx`	IN/OUT: Index of the property to begin with
H5P_iterate_t `iter_func`	IN: Function pointer to function to be called with each property iterated over
void * `iter_data`	IN/OUT: Pointer to iteration data from user

Description:

H5P_ITERATE iterates over the properties in the property object specified in `id`, which may be either a property list or a property class, performing a specified operation on each property in turn.

For each property in the object, `iter_func` and the additional information specified below are passed to the `H5P_iterate_t` operator function.

The iteration begins with the `idx`-th property in the object; the next element to be processed by the operator is returned in `idx`. If `idx` is NULL, the iterator starts at the first property; since no stopping point is returned in this case, the iterator cannot be restarted if one of the calls to its operator returns non-zero.

The prototype for the `H5P_iterate_t` operator is as follows:

```
typedef herr_t (*H5P_iterate_t)( hid_t id, const char *name, void *iter_data )
```

The operation receives the property list or class identifier for the object being iterated over, `id`, the name of the current property within the object, `name`, and the pointer to the operator data passed in to H5P_ITERATE, `iter_data`. The valid return values from an operator are as follows:

Zero	Causes the iterator to continue, returning zero when all properties have been processed
Positive	Causes the iterator to immediately return that positive value, indicating short-circuit success. The iterator can be restarted at the index of the next property
Negative	Causes the iterator to immediately return that value, indicating failure. The iterator can be restarted at the index of the next property

H5P_ITERATE assumes that the properties in the object identified by `id` remain unchanged through the iteration. If the membership changes during the iteration, the function's behavior is undefined.

Programming Note for C++ Developers Using C Functions:

If a C routine that takes a function pointer as an argument is called from within C++ code, the C routine should be returned from normally.

Examples of this kind of routine include callbacks such as H5P_SET_ELINK_CB and H5P_SET_TYPE_CONV_CB and functions such as H5T_CONVERT and H5E_WALK2.

Exiting the routine in its normal fashion allows the HDF5 C library to clean up its work properly. In other words, if the C++ application jumps out of the routine back to the C++ “catch” statement, the library is not given the opportunity to close any temporary data structures that were set up when the routine was called. The C++ application should save some state as the routine is started so that any problem that occurs might be diagnosed.

Returns:

Success: the return value of the last call to iter_func if it was non-zero; zero if all properties have been processed

Failure: a negative value

Example:

History:

--- Last Modified: August 12, 2019 | 12:12 PM