HDF5 REST VOL Connector User's Guide

Jordan Henderson

This document aims to be a helpful guide on how to use the HDF5 REST VOL connector to leverage the capabilities of the HDF Kita cloud-based storage system within an HDF5 application.

Revision History

Version Number	Date	Comments
v0.1	Apr. 3, 2020	First draft.

Contents

List of Figures

1. Overview

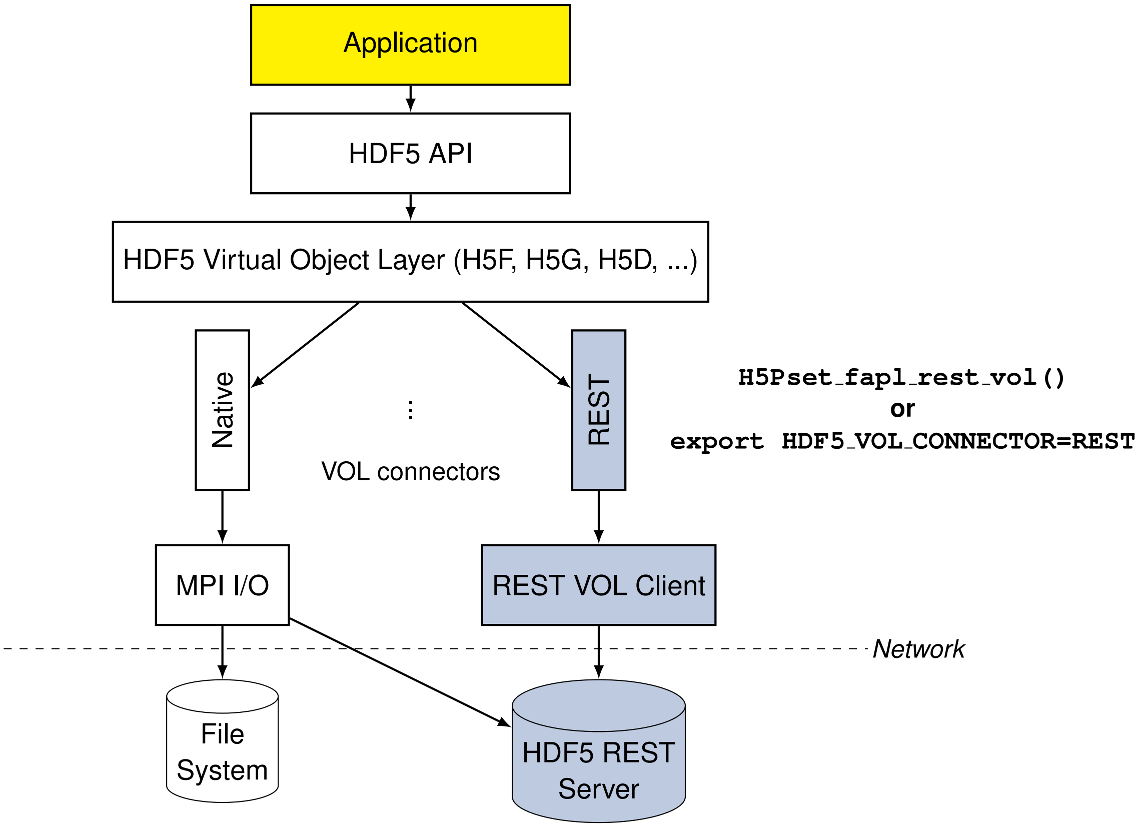

Figure 1 – within Virtual Object Layer. All of the HDF5 I/O related calls are routed to the VOL connector.

2. Using the REST VOL connector within an HDF5 application

This section outlines the unique aspects of writing, building and running HDF5 applications with the VOL connector.

2.1. Building the HDF5 REST VOL connector

TODO - refer to README.md

2.2. Writing HDF5 REST VOL connector applications

There are currently two main ways to tell an existing HDF5 application to use the VOL connector: either *implicitly* by using environment variables to tell the HDF5 library to load the connector as a dynamically loaded plugin or *explicitly* by making use of HDF5 property lists.

2.2.1. With the REST VOL connector as a dynamically-loaded plugin

HDF5 has the capability to dynamically load and use a VOL connector for running applications with. In order to choose a particular VOL connector to use, two initial steps must be taken. First, one must help HDF5 locate the VOL connector by pointing to the directory which contains the built library. This can be accomplished by setting the environment variable `HDF5_PLUGIN_PATH` to this directory. Next, HDF5 needs to know the name of which library to use, which is configured by setting the environment variable `HDF5_VOL_CONNECTOR` to the name of the connector.

In order to use the VOL connector, the aforementioned environment variables should be set as:

```
HDF5_PLUGIN_PATH=/rest/vol/installation/directory/lib
HDF5_VOL_CONNECTOR=REST
```

Having completed this step, HDF5 will be setup to load the VOL connector and use it for running applications, including HDF5's own tests. No additional modifications will need to be made to the existing HDF5 application.

2.2.2. Without the REST VOL connector as a dynamically-loaded plugin

If dynamic loading of the VOL connector is not used, any HDF5 application using the connector must:

1. Include `rest_vol_public.h`, found in the `include` directory of the VOL connector installation directory.
2. Link against `libhdf5_vol_rest.so` (or similar), found in the `lib` directory of the VOL connector installation directory. Note that dependencies can alternatively be retrieved through CMake or pkg-config.

In this particular case, an HDF5 VOL connector application will also require three new function calls in addition to those for an equivalent HDF5 application (see Appendix A for more details):

- `H5rest_init()` — Initializes the VOL connector

  Called upon application startup, before any file is accessed.

- `H5Pset_fapl_rest_vol()` — Sets VOL connector access on a File Access Property List.

  Called to prepare a FAPL to open a file through the VOL connector. See HDF5 File Access Property Lists for more information about File Access Property Lists.

- `H5rest_term()` — Cleanly shutdowns the VOL connector

  Called on application shutdown, after all files have been closed.

2.2.3. Skeleton Example

Below is a no-op application that opens and closes a file using the VOL connector. For clarity, no error-checking is performed. Note that this example is meant only for the case when the VOL connector is not being dynamically loaded.

```
#include "hdf5.h"
#include "rest_vol_public.h"

int main(void)
{
    hid_t file_id;
    hid_t fapl_id;

    /* Initialize REST VOL connector */
    H5rest_init();

    fapl_id = H5Pcreate(H5P_FILE_ACCESS);
    H5Pset_fapl_rest_vol(fapl_id);

    file_id = H5Fopen("my_file.h5", H5F_ACC_RDWR, fapl_id);

    /* Operate on file */
    [...]

    H5Pclose(fapl_id);
    H5Fclose(file_id);

    /* Terminate the REST VOL connector. */
    H5rest_term();

    return 0;
}
```

2.3. Building HDF5 REST VOL connector applications

Assuming an HDF5 application has been written following the instructions in the previous section, the application should be built as normal for any other HDF5 application. However, if the VOL connector is not being dynamically loaded, the steps in the following section are required to build the application.

2.3.1. Without the REST VOL connector as a dynamically-loaded plugin

To link in the required libraries, the compiler will likely require the additional linker flags:

```
-lhdf5_vol_rest -lcurl -lyajl
```

However, these flags may vary depending on platform, compiler and installation location of the VOL connector. It is highly recommended that compilation of HDF5 VOL connector applications be done using either the `h5cc` script included with HDF5 distributions, or CMake/ pkg-config, as these will manage linking with the HDF5 library. The above notice about additional library linking applies to usage of `h5cc/h5pcc`. For example:

```
h5cc -lhdf5_vol_rest -lcurl -lyajl my_application.c -o my_application
```

2.4. Running HDF5 REST VOL connector applications

2.4.1. Server Access

Running applications that use the VOL connector connector requires access to a running HDF Kita server. Refer to HDF Kita for more information on the setup process for this.

2.4.2. Connection Information

For the VOL connector to correctly interact with a running HDF Kita server instance, the connector must be passed the base URL of the HDF Kita endpoint, as well as any authentication credential information needed. This can be accomplished in one of the following ways:

Environment Variables

- `HSDS_USERNAME` - (optional) The username to use for authentication
- `HSDS_PASSWORD` - (optional) The password to use for authentication
- `HSDS_ENDPOINT` - The base URL of the HDF Kita instance (e.g. http://hsdshdflab.hdfgroup.org)

Configuration File

TODO

Note that there are cases where authentication may not be required, such as when simply retrieving information from a publicly-accessible HDF5 dataset or similar. In these cases, it is only necessary to supply the HDF Kita endpoint via the HSDS_ENDPOINT environment variable or the .hscfg configuration file.

2.4.3. Example Applications

Some of the example C applications which are included with HDF5 distributions have been adapted to work with the VOL connector and are included under the top-level `examples` directory in the VOL connector source root directory. The built example applications can be run from the `bin` directory inside the build directory.

In addition to these examples, the `test/vol-tests` directory contains several test files, each containing test functions that are examples of HDF5 applications in miniature, focused on a particular behavior. These mini-application tests cover a moderate amount of HDF5's public API functionality and should be a good indicator of whether the VOL connector is working correctly in conjunction with a running HDF Kita instance. Note that these tests currently rely on HDF5's dynamically-loaded VOL connector capabilities in order to run with the VOL connector.

3. HDF5 API Support

3.1. Feature Specific Support

The following sections serve to illustrate the VOL connector's support for features in HDF5, as well as to highlight any differences between the expected behavior of an HDF5 feature versus the actual behavior as implemented by the VOL connector.

3.1.1. Attribute Features

Feature			Supported?	Notes
Dataspace	Dimensionality	H5S_NULL	Yes	
		H5S_SCALAR	Yes	
		SIMPLE	Yes	
Datatype		Atomic	Yes	Non-predefined integer and floating-point datatypes are currently not supported.
		Compound	Yes	
		Variable-length	No	
		Array	Yes	
		String	Yes	Only `H5T_STR_NULLPAD` is supported for string padding for fixed-length strings. Only `H5T_STR_NULLTERM` is supported for string padding for variable-length strings.
		Enum	Yes	
		Opaque	No	
		Bitfield	No	
		Time	No	
		Reference	No	

3.1.2. Dataset Features

Feature			Supported?	Notes
Dataspace	Dimensionality	H5S_NULL	Yes	
		H5S_SCALAR	Yes	
		SIMPLE	Yes	
	Selection Type	NONE	Yes	Non-regular and non-contiguous hyperslabs are currently not supported.
		H5S_ALL	Yes	
		Hyperslab Selection	Yes	
		Point Selection	Yes	
Datatype		Atomic	Yes	Non-predefined integer and floating-point datatypes are currently not supported.
		Compound	Yes	
		Variable-length	No	
		Array	Yes	
		String	Yes	Only `H5T_STR_NULLPAD` is supported for string padding for fixed-length strings. Only `H5T_STR_NULLTERM` is supported for string padding for variable-length strings.
		Enum	Yes	

Feature			Supported?	Notes
Properties	Storage Properties (creation)	Compact	No	Setting is ignored; stored as contiguous.
		External	No	Setting is ignored; stored as contiguous.
		Contiguous	Yes	
		Chunked	Yes	
		VDS	No	The VDS feature is not currently planned to be supported.
	Other Properties (creation)	Attribute Creation Order	Yes	
		Fill Value	No	
		Filters	No	HDF5 does not expose any public APIs for working with the filter pipeline; however, this feature may be supported in the future.
		Storage Allocation Time	N/A	

Feature			Supported?	Notes
Properties (cont.)	Access Properties	Chunk cache	No	HDF5 does not expose any public APIs for implementing a chunk cache for arbitrary VOL connectors; however, this feature may be supported in the future.
		VDS views and printf	No	The VDS feature is not currently planned to be supported.
		MPI-I/O Collective Metadata Ops	No	
	Transfer Properties	MPI-I/O Independent or Collective I/O mode	N/A	

3.1.3. File Features

Feature			Supported?	Notes
File creation flags		H5F_ACC_TRUNC	Yes	The file creation flags behave as for native HDF5.
		H5F_ACC_EXCL	No	
File opening flags		H5F_ACC_RDWR	Yes	
		H5F_ACC_RDONLY	Yes	
Properties	Creation Properties	Attribute Creation Order	Yes	The rest of the file creation properties are related to the native HDF5-specific file format.
	Access Properties (Drivers)	SEC2 Driver	N/A	These drivers are applicable to native HDF5 only.
		Family Driver	N/A	
		Split Driver	N/A	
		Multi Driver	N/A	
		Core Driver	N/A	
		Log Driver	N/A	
		MPI-I/O	No	This feature may be supported in the future.

Feature			Supported?	Notes
Properties (cont.)	Access Properties (Other)	MPI-I/O Collective Metadata Ops	No	
		User block	N/A	
		Chunk Cache	No	HDF5 does not expose any public APIs for implementing a chunk cache for arbitrary VOL connectors; however, this feature may be supported in the future.
		Object flushing callbacks	N/A	
		File closing degree	N/A	
		Evict on close	N/A	
		Sieve buffer size for partial I/O	No	HDF5 does not expose any public APIs for implementing a partial I/O sieve buffer for arbitrary VOL connectors; however, this feature may be supported in the future.
		File Image	N/A	

3.1.4. Group Features

Feature			Supported?	Notes
Properties	Creation Properties	Link Creation Order	Yes	
		Attribute Creation Order	Yes	
		Other Properties	N/A	These properties are related to the native HDF5-specific file format.
	Access Properties	MPI-I/O Collective Metadata Ops	No	

3.2. API Specific Support

The following sections serve to illustrate the VOL connector's support for the HDF5 API, as well as to highlight any differences between the expected behavior of an HDF5 API call versus the actual behavior as implemented by the VOL connector. If a particular HDF5 API call does not appear among these tables, it is most likely a native HDF5-specific API call which cannot be implemented by non-native HDF5 VOL connectors. These types of API calls are listed among the tables in Appendix B.

3.2.1. H5A interface

Supported API calls

API call	Notes
H5Acreate(1/2)	
H5Acreate_by_name	
H5Aopen(_by_name)	
H5Aopen_name	Deprecated in favor of H5A_open_by_name
H5Awrite	
H5Aread	
H5Aclose	
H5Aiterate(2)	
H5Aiterate_by_name	
H5Aexists(_by_name)	
H5Adelete(_by_name)	
H5Aget_name	
H5Aget_space	
H5Aget_type	
H5Aget_info(_by_name)	Of the four fields in the `H5A_info_t` struct: ▪ `corder_valid` is currently alwasy set to FALSE ▪ `corder` is currently always set to 0 ▪ `cset` is currently always set to `H5T_CSET_ASCII` ▪ `data_size` is currently always set to 0
H5Aget_create_plist	

Currently unsupported API calls

API call	Notes
H5Aopen_by_idx	
H5Aopen_idx	Deprecated in favor of H5Aopen_by_idx
H5Aget_name_by_idx	
H5Aget_info_by_idx	
H5Aget_storage_size	
H5Adelete_by_idx	
H5Arename(_by_name)	

3.2.2. H5D interface

Supported API calls

API call	Notes
H5Dcreate(1/2)	
H5Dcreate_anon	
H5Dopen(1/2)	
H5Dwrite	
H5Dread	
H5Dclose	
H5Dget_space	
H5Dget_type	
H5Dget_create_plist	
H5Dget_access_plist	

Currently unsupported API calls

API call	Notes
H5Dget_space_status	
H5Dget_storage_size	
H5Dextend	
H5Dset_extent	
H5Dflush	
H5Drefresh	

3.2.3. H5F interface

Supported API calls

API call	Notes
H5Fcreate	
H5Fopen	
H5Freopen	
H5Fget_create_plist	
H5Fget_access_plist	
H5Fget_intent	
H5Fget_name	
H5Fclose	

Currently unsupported API calls

API call	Notes
H5Fis_accessible	
H5Fflush	
H5Fmount	
H5Funmount	
H5Fdelete	
H5Fget_obj_count	
H5Fget_obj_ids	

3.2.4. H5G interface

Supported API calls

API call	Notes
H5Gcreate(1/2)	
H5Gcreate_anon	
H5Gopen(1/2)	
H5Gclose	
H5Gunlink	
H5Gget_create_plist	
H5Gget_info(_by_name)	Of the four fields in the `H5G_info_t` struct: ■ `storage_type` is currently always set to `H5G_STORAGE_TYPE_SYMBOL_TABLE` ■ `nlinks` is set appropriately ■ `max_corder` is currently always set to 0 ■ `mounted` is currently always set to `FALSE`
H5Gget_linkval	
H5Gget_num_objs	
H5Glink(2)	Currently only hard and soft link creation are supported.
H5Gmove(2)	Refer to Notes for `H5Lmove`

Currently unsupported API calls

API call	Notes
H5Gget_info_by_idx	
H5Gget_objname_by_idx	
H5Gflush	
H5Grefresh	

3.2.5. H5L interface

Supported API calls

API call	Notes
H5Lcreate_hard	Reference count tracking is not currently implemented, so objects will not be removed when the last hard link pointing to them is removed
H5Lcreate_soft	
H5Lexists	
H5Literate(_by_name)	
H5Lvisit(_by_name)	
H5Ldelete	Reference count tracking is not currently implemented, so objects will not be removed when the last hard link pointing to them is removed

API call	Notes
H5Lget_info	Of the five fields in the `H5L_info_t` struct: ▪ `type` is set appropriately ▪ `corder_valid` is currently always set to FALSE ▪ `corder` is currently always set to 0 ▪ `cset` is currently always set to `H5T_CSET_ASCII` ▪ `u` has member `address` or `val_size` set appropriately based on whether the link is a hard link or not
H5Lget_val	

Currently unsupported API calls

API call	Notes
H5Lcreate_external	
H5Lcreate_ud	
H5Lget_info_by_idx	
H5Lget_val_by_idx	
H5Lget_name_by_idx	
H5Ldelete_by_idx	
H5Lcopy	
H5Lmove	

3.2.6. H5O interface

Supported API calls

API call	Notes
H5Oopen	
H5Oclose	
H5Olink	

Currently unsupported API calls

API call	Notes
H5Oopen_by_token	
H5Oopen_by_idx	
H5Oget_info(_by_name/_by_idx)	
H5Oincr_refcount	
H5Odecr_refcount	
H5Oexists_by_name	
H5Ovisit(1/2)	
H5Ovisit_by_name(1/2)	
H5Ocopy	
H5Oflush	
H5Orefresh	

3.2.7. H5R interface

Supported API calls

API call	Notes
	[1]

Currently unsupported API calls

API call	Notes
H5Rcreate_object	
H5Rcreate_region	
H5Rcreate_attr	
H5Ropen_object	
H5Ropen_region	
H5Ropen_attr	
H5Rget_obj_type3	
H5Rget_file_name	
H5Rget_obj_name	

[1] The REST VOL connector has not yet been updated to use HDF5's new reference API.

3.2.8. H5T interface

Supported API calls

API call	Notes
H5Tcommit(1/2)	
H5Tcommit_anon	
H5Topen(1/2)	
H5Tclose	
H5Tget_create_plist	

Currently unsupported API calls

API call	Notes
H5Tflush	
H5Trefresh	

3.3. Known Limitations

The following outlines the known current limitations of the VOL connector.

- Trying to open an object in a file by using a pathname where any component of the path, except for the last component, is a soft link is currently not supported. For example, trying to open a dataset by the pathname '/group/subgroup/soft_link_to_dataset' should work. However, trying to open a dataset by the pathname '/group/soft_link_to_group/soft_link_to_dataset' will generally fail.
- Using a trailing '/' character on path names will currently cause problems with the connector and result in incorrect behavior.
- The use of HDF5 point selections for dataset writes will generally incur an additional memory overhead of approximately 4/3 the size of the original buffer used for the `H5Dwrite` call. This is due to needing a temporary copy of the buffer which is base64-encoded for the server transfer.

4. Testing the REST VOL connector

The following sections cover how to test the VOL connector, as well as the individual components of the VOL connector's overall testing infrastructure.

4.1. With Autotools

TODO

4.2. With CMake

Once the VOL connector has been built, running the connector's tests should be as simple as running

```
ctest .
```

from the build directory. This will run each of the connector's test components in turn. For more information on using CTest's options to control testing behavior, refer to the CTest Documentation.

4.3. REST VOL connector's testing components

4.3.1. Generic HDF5 VOL connector test suite

In order to test HDF5 VOL connectors to make sure that they are functioning as expected, a suite of tests which only use the public HDF5 API has been written. This suite of tests is available under the path:

`test/vol-tests`

and when built, will appear as the `h5vl_test` and `h5vl_test_parallel` executables in the `bin` directory inside the build directory.

Note that running this test suite requires that your environment is setup to have HDF5 dynamically load the VOL connector. Also, this test suite currently does not have the capability to query what kind of functionality an HDF5 VOL connector supports and therefore certain tests will be skipped if they use an HDF5 API call which is not implemented, or which is specifically unsupported, by the VOL connector.

4.3.2. REST VOL connector-specific test suite

In addition to the generic VOL connector testing suite, the VOL connector also includes the following test suites, which test features specific to the connector:

TODO

A. Reference Manual

A.1. H5rest_init

Synopsis:

```
herr_t H5rest_init(void);
```

Purpose:

Initialize the REST VOL connector.

Description:

H5rest_init initializes the VOL connector by registering the connector with the library, initializing libcurl, and then performing other internal setup.

Parameters:

None.

Returns:

Returns a non-negative value if successful; otherwise returns a negative value.

A.2. H5rest_term

Synopsis:

```
herr_t H5rest_term(void);
```

Purpose:

Terminate the REST VOL connector.

Description:

`H5rest_term` terminates the REST VOL connector.

Parameters:

None.

Returns:

Returns a non-negative value if successful; otherwise returns a negative value.

A.3. H5Pset_fapl_rest_vol

Synopsis:

```
herr_t H5Pset_fapl_rest_vol(hid_t fapl_id);
```

Purpose:

Set the file access property list to use the REST VOL connector.

Description:

`H5Pset_fapl_rest_vol` modifies the file access property list to use the REST VOL connector.

Parameters:

`hid_t fapl_id`	IN: File access property list ID

Returns:

Returns a non-negative value if successful; otherwise returns a negative value.

B. Native HDF5 VOL connector-specific API calls

The following HDF5 API calls are either specific to the native HDF5 VOL connector or are not routed through the VOL and thus are not able to be implemented by the VOL connector (or other VOL connectors):

B.1. H5A interface

API call	Notes
H5Aiterate1	Deprecated in favor of H5Aiterate2
H5Aget_num_attrs	Deprecated in favor of H5Oget_info

B.2. H5D interface

API call	Notes
H5Dformat_convert	
H5Dget_offset	
H5Dget_chunk_index_type	
H5Dget_chunk_storage_size	
H5Dvlen_reclaim	
H5Dvlen_get_buf_size	
H5Diterate	
H5Dscatter	
H5Dgather	
H5Dfill	
H5Dread_chunk	
H5Dwrite_chunk	

B.3. H5F interface

API call	Notes
H5Fis_hdf5	Uses a default FAPL so can only ever be routed through the native HDF5 VOL connector
H5Fget_vfd_handle	
H5Fget_freespace	
H5Fget_filesize	
H5Fget_file_image	
H5Fget_mdc_config	
H5Fset_mdc_config	
H5Fget_mdc_hit_rate	
H5Fget_mdc_size	
H5Freset_mdc_hit_rate_stats	
H5Fget_info(1/2)	
H5Fget_metadata_read_retry_info	
H5Fget_free_sections	
H5Fclear_elink_file_cache	
H5Fstart_swmr_write	
H5Fstart_mdc_logging	
H5Fstop_mdc_logging	
H5Fget_mdc_logging_status	
H5Fset_libver_bounds	
H5Fformat_convert	
H5Freset_page_buffering_stats	
H5Fget_page_buffering_stats	
H5Fget_mdc_image_info	
H5Fget_eoa	
H5Fincrement_filesize	
H5Fget_dset_no_attrs_hint	
H5Fset_dset_no_attrs_hint	
H5Fset_latest_format	
H5Fset_mpi_atomicity	
H5Fget_mpi_atomicity	

B.4. H5G interface

API call	Notes
H5Gset_comment	Deprecated in favor of H5Oset_comment/H5Oset_comment_by_name
H5Gget_comment	Deprecated in favor of H5Oget_comment/H5Oget_comment_by_name
H5Giterate	Deprecated in favor of H5Literate
H5Gget_objinfo	Deprecated in favor of H5Lget_info/H5Oget_info
H5Gget_objtype_by_idx	Deprecated in favor of H5Lget_info/H5Oget_info

B.5. H5L interface

API call	Notes
H5Lregister	
H5Lunregister	
H5Lis_registered	
H5Lunpack_elink_val	

B.6. H5O interface

API call	Notes
H5Oget_info(1/2)	
H5Oget_info_by_name(1/2)	
H5Oget_info_by_idx(1/2)	
H5Oset_comment(_by_name)	Deprecated in favor of using attributes on objects
H5Oget_comment(_by_name)	Deprecated in favor of using attributes on objects
H5Oare_mdc_flushes_disabled	
H5Oenable_mdc_flushes	
H5Odisable_mdc_flushes	

B.7. H5R interface

API call	Notes
H5Rdestroy	
H5Rget_type	
H5Requal	
H5Rcopy	
H5Rget_attr_name	

B.8. H5T interface

API call	Notes