

---

# H4toH5 Conversion Library API Reference Manual

---

*with*  
H4/H5 Command-line Conversion Utilities

---

---

Release 2.2.3  
September 2017

The HDF Group

<https://www.hdfgroup.org/>

---

---

# Table of Contents

Overview	3
<hr/>	
H4/H5 Command-line Conversion Utilities	
<hr/>	
h4toh5	6
h5toh4	9
<hr/>	
H4toH5 Conversion Library APIs	
<hr/>	
General functions	13
Global attributes	17
Datatypes	19
Dimension scales	21
Annotations	25
Error messages	33
Vgroups	35
Vdatas	39
SDSs	45
Images and palettes	49
Lone objects	55

---

Relevant HDF Group links:  
[HDF Group Home Page](#)  
[HDF4/HDF5 Conversion Tools](#)

H4toH5 Conversion Library documents:  
[Reference Manual \(PDF\)](#)  
[User's Guide \(PDF\)](#)  
[Programmer's Notes \(PDF\)](#)  
[Mapping HDF4 Objects to HDF5 Objects \(PDF\)](#)

---

# H4/H5 Command-line Conversion Utilities

## h4toh5 and h5toh4

These convenience utilities can be run from the UNIX or Linux command line or used in scripts to perform generic HDF4-to-HDF5 or HDF5-to-HDF4 file conversions.

### Command-line utilities

[h4toh5](#)

[h5toh4](#)

# H4toH5 Conversion Library

## API Reference Manual

The H4toH5 Conversion Library, generally referred to as the H4toH5 library, provides functions for the custom conversion of HDF4 data structures and objects to corresponding HDF5 data structures and objects.

### General functions

[H4toH5open](#)  
[H4toH5open\\_id](#)  
[H4toH5close](#)

### Global attributes

[H4toH5glo\\_sds\\_attr](#)  
[H4toH5glo\\_image\\_attr](#)

### Datatypes

[H4toH5datatype](#)

### Dimension scales

[H4toH5one\\_dimscale](#)  
[H4toH5all\\_dimscale](#)

### Annotations

[H4toH5anno\\_file\\_label](#)  
[H4toH5anno\\_file\\_desc](#)  
[H4toH5anno\\_file\\_all\\_labels](#)  
[H4toH5anno\\_file\\_all\\_descs](#)  
[H4toH5anno\\_obj\\_label](#)  
[H4toH5anno\\_obj\\_desc](#)

### Vgroups

[H4toH5bas\\_vgroup](#)  
[H4toH5vgroup\\_attr\\_name](#)  
[H4toH5vgroup\\_attr\\_index](#)  
[H4toH5adv\\_group](#)

### Vdatas

[H4toH5vdata](#)  
[H4toH5vdata\\_attr\\_name](#)  
[H4toH5vdata\\_attr\\_index](#)  
[H4toH5vdata\\_field\\_attr\\_name](#)  
[H4toH5vdata\\_field\\_attr\\_index](#)

### SDSs

[H4toH5sds](#)  
[H4toH5sds\\_attr\\_name](#)  
[H4toH5sds\\_attr\\_index](#)

### Images and palettes

[H4toH5image](#)  
[H4toH5image\\_attr\\_name](#)  
[H4toH5image\\_attr\\_index](#)  
[H4toH5pal](#)

H4toH5anno\_obj\_all\_labels  
H4toH5anno\_obj\_all\_descs

**Error messages**

H4toH5error\_get

**Lone objects**

H4toH5all\_lone\_sds  
H4toH5all\_lone\_image  
H4toH5all\_lone\_vdata

## H4toH5 Conversion Library Reference Manual

### *Alphabetical Listing*

<a href="#"><u>H4toH5adv_group</u></a>	<a href="#"><u>H4toH5anno_obj_label</u></a>	<a href="#"><u>H4toH5open_id</u></a>
<a href="#"><u>H4toH5all_dimscale</u></a>	<a href="#"><u>H4toH5bas_vgroup</u></a>	<a href="#"><u>H4toH5pal</u></a>
<a href="#"><u>H4toH5all_lone_image</u></a>	<a href="#"><u>H4toH5close</u></a>	<a href="#"><u>H4toH5sds</u></a>
<a href="#"><u>H4toH5all_lone_sds</u></a>	<a href="#"><u>H4toH5datatype</u></a>	<a href="#"><u>H4toH5sds_attr_index</u></a>
<a href="#"><u>H4toH5all_lone_vdata</u></a>	<a href="#"><u>H4toH5error_get</u></a>	<a href="#"><u>H4toH5sds_attr_name</u></a>
<a href="#"><u>H4toH5anno_file_all_descs</u></a>	<a href="#"><u>H4toH5glo_image_attr</u></a>	<a href="#"><u>H4toH5vdata</u></a>
<a href="#"><u>H4toH5anno_file_all_labels</u></a>	<a href="#"><u>H4toH5glo_sds_attr</u></a>	<a href="#"><u>H4toH5vdata_attr_index</u></a>
<a href="#"><u>H4toH5anno_file_desc</u></a>	<a href="#"><u>H4toH5image</u></a>	<a href="#"><u>H4toH5vdata_attr_name</u></a>
<a href="#"><u>H4toH5anno_file_label</u></a>	<a href="#"><u>H4toH5image_attr_index</u></a>	<a href="#"><u>H4toH5vdata_field_attr_index</u></a>
<a href="#"><u>H4toH5anno_obj_all_descs</u></a>	<a href="#"><u>H4toH5image_attr_name</u></a>	<a href="#"><u>H4toH5vdata_field_attr_name</u></a>
<a href="#"><u>H4toH5anno_obj_all_labels</u></a>	<a href="#"><u>H4toH5one_dimscale</u></a>	<a href="#"><u>H4toH5vgroup_attr_index</u></a>
<a href="#"><u>H4toH5anno_obj_desc</u></a>	<a href="#"><u>H4toH5open</u></a>	<a href="#"><u>H4toH5vgroup_attr_name</u></a>

---



---

## H4/H5 Command-line Conversion Utilites

---

**Tool Name:** h4toh5

**Syntax:**

```
h4toh5 -h
h4toh5 h4file h5file
h4toh5 h4file
```

**Purpose:**

Converts an HDF4 file to an HDF5 file.

**Description:**

`h4toh5` is a file conversion utility that reads an HDF4 file, *h4file* (`input.hdf` for example), and writes an HDF5 file, *h5file* (`output.h5` for example), containing the same data.

If no output file *h5file* is specified, `h4toh5` uses the input filename to designate the output file, replacing the extension `.hdf` with `.h5`. For example, if the input file `scheme3.hdf` is specified with no output filename, `h4toh5` will name the output file `scheme3.h5`.

The `-h` option causes a syntax summary similar to the following to be displayed:

```
h4toh5 inputfile.hdf outputfile.h5
h4toh5 inputfile.hdf
h4toh5 -pa inputfile.hdf outputfile.h5
h4toh5 -ospe inputfile.hdf outputfile.h5
h4toh5 -osna inputfile.hdf outputfile.h5
h4toh5 -na inputfile.hdf outputfile.h5
h4toh5 -sv inputfile.hdf outputfile.h5
```

`h4toh5` options can be described as the following:

Option	Description
<code>h4toh5</code>	Use the HDF5 dimension scale APIs, Not add the H4toH5 predefined attributes.
<code>h4toh5 -pa</code>	Use the HDF5 dimension scale APIs, Not add the H4toH5 predefined attributes.
<code>h4toh5 -ospe</code>	Follow the older version(version 3 and before) of the H4toH5 mapping document.
<code>h4toh5 -na</code> or <code>h4toh5 -osna</code>	Follow the older version of the mapping document, Not add the H4toH5 predefined attributes.
<code>h4toh5 -sv</code>	Map one field <code>vdata</code> to an HDF5 atomic datatype dataset. Use the HDF5 dimension scale APIs. Not add the H4toH5 predefined attributes.

Each object in the HDF4 file is converted to an equivalent HDF5 object, according to the mapping described in [Mapping HDF4 Objects to HDF5 Objects](#).

h4toh5 converts the following HDF4 objects:

<b>HDF4 Object</b>	<b>Resulting HDF5 Object</b>
SDS	Dataset
GR, RI8, and RI24 image	Dataset
Vdata	Dataset
Vgroup	Group
Annotation	Attribute
Palette	Dataset

**HDF-EOS2 to netCDF-4 file conversion:** h4toh5 can also be built and installed with a `--with-hdfeos2` option. This option provides the capacity to convert an HDF-EOS2 file to an HDF5 file that can be accessed by netCDF-4. Instructions for building and using this feature appear in the files [RELEASE.txt](#) and [INSTALL\\_Unix.txt](#) in the distributed “h4toh5 Conversion Software” source code package. These files are also available here in the distributed source code:

```
<source_code>/release_docs/RELEASE.txt  
<source_code>/release_docs/INSTALL_Unix.txt
```

**Options and Parameters:**

- h        Displays a syntax summary.
- h4file*    The HDF4 file to be converted.
- h5file*    The HDF5 file to be created.

**Exit Status:**

- 0        Succeeded.
- > 0     An error occurred.



**Tool Name:** h5toh4**Syntax:**

```
h5toh4 -h
h5toh4 h5file h4file
h5toh4 h5file
h5toh4 -m h5file1 h5file2 h5file3 ...
```

**Purpose:**

Converts an HDF5 file into an HDF4 file.

**Description:**

`h5toh4` is an HDF5 utility which reads an HDF5 file, *h5file*, and converts all supported objects and pathways to produce an HDF4 file, *h4file*. If *h4file* already exists, it will be replaced.

If only one file name is given, the name must end in `.h5` and is assumed to represent the HDF5 input file. `h5toh4` replaces the `.h5` suffix with `.hdf` to form the name of the resulting HDF4 file and proceeds as above. If a file with the name of the intended HDF4 file already exists, `h5toh4` exits with an error without changing the contents of any file.

The `-m` option allows multiple HDF5 file arguments. Each file name is treated the same as the single file name case above.

The `-h` option causes the following syntax summary to be displayed:

```
h5toh4 file.h5 file.hdf
h5toh4 file.h5
h5toh4 -m file1.h5 file2.h5 ...
```

The following HDF5 objects occurring in an HDF5 file are converted to HDF4 objects in the HDF4 file:

- ◇ HDF5 group objects are converted into HDF4 Vgroup objects. HDF5 hard links and soft links pointing to objects are converted to HDF4 Vgroup references.
- ◇ HDF5 dataset objects of integer datatype are converted into HDF4 SDS objects. These datasets may have up to 32 fixed dimensions. The slowest varying dimension may be extendable. 8-bit, 16-bit, and 32-bit integer datatypes are supported.
- ◇ HDF5 dataset objects of floating point datatype are converted into HDF4 SDS objects. These datasets may have up to 32 fixed dimensions. The slowest varying dimension may be extendable. 32-bit and 64-bit floating point datatypes are supported.
- ◇ HDF5 dataset objects of single dimension and compound datatype are converted into HDF4 Vdata objects. The length of that single dimension may be fixed or extendable. The members of the compound datatype are constrained to be no more than rank 4.
- ◇ HDF5 dataset objects of single dimension and fixed-length string datatype are converted into HDF4 Vdata objects. The HDF4 Vdata is a single field whose order is the length of the HDF5 string type. The number of records of the Vdata is the length of the single dimension which may be fixed or extendable.

Other objects are not converted and are not recorded in the resulting *h4file*.

Attributes associated with any of the supported HDF5 objects are carried over to the HDF4 objects. Attributes may be of integer, floating point, or fixed length string datatype and they may have up to 32 fixed dimensions.

All datatypes are converted to big-endian. Floating point datatypes are converted to IEEE format.

### Limitations:

`h5toh4` performs a basic conversion designed to meet common requirements; it does not attempt to convert every HDF5 object that a file might contain.

This section describes some of the tool's limitations:

#### Limited datatypes conversions

`h5toh4` does not convert datasets of the following datatypes:

- ◆ Array
- ◆ Multi-dimensional strings
- ◆ Bitfield
- ◆ Time

The tool issues a warning message and skips the dataset, but continues to process the file.

#### Unrecognized datatypes

`h5toh4` issues an error and exits upon encountering a dataset of the following datatypes:

- ◆ Object reference
- ◆ Region reference
- ◆ ENUM
- ◆ Any variable-length datatype

#### Compound datatype

When `h5toh4` encounters a dataset of the following datatype, it causes a segmentation fault:

- ◆ Compound datatype with a field of array datatype and 5 or more dimensions

The tool successfully processes compound datatypes with fields of array datatype and 4 or fewer dimensions.

#### Dimension scales

`h5toh4` does not support dimension scale datasets; they are treated as regular datasets in HDF5 and SDSs in the conversion to HDF4.

#### Unlimited dimensions

HDF4 allows only one unlimited dimension, so `h5toh4` is limited in its ability to convert HDF5 datasets with unlimited dimensions.

If the first dimension of an HDF5 dataset is unlimited, that is if

`dim[0]=H5S_UNLIMITED`, then `h5toh4` will set that dimension to unlimited in the resulting HDF4 dataset. Other dimensions will be set in the HDF4 file to their current size in the HDF5 file.

#### Number of objects

An HDF4 file is limited to 20,000 objects. `h5toh4` cannot support file conversions that would exceed that limit (and its behavior in such situations is not tested).

**File size**

`h5toh4` can convert files up to 2 GB in size; it will exit with code 1 if the source HDF5 file is larger than 2 GB.

**Options and Parameters:**

- h        Displays a syntax summary.
- m        Converts multiple HDF5 files to multiple HDF4 files.
- h5file*    The HDF5 file to be converted.
- h4file*    The HDF4 file to be created.

**Exit Status:**

- 0        Succeeded.
- > 0     An error occurred.



---



---

## H4toH5 Conversion Library APIs

---



---



---

### General functions

---

**Name:** `H4toH5open`

**Signature:**

`hid_t H4toH5open(char * H4file_name, char * H5file_name, int access)`

**Purpose:**

Initializes the H4toH5 library interface.

**Description:**

`H4toH5open` initializes the H4toH5 library to convert data from one HDF4 file to one HDF5 file. Either this function or `H4toH5open_id` must be the first H4toH5 library function called in an application.

If `H5file_name` is set to `NULL`, the HDF5 filename will be similar to the HDF4 filename, but with the extension `.h5`.

The access flag specifies whether an existing HDF5 file is to be overwritten.

The `h4toh5` identifier returned by this function must be released with `H4toH5close` or resource leaks will develop.

**Parameters:**

<code>char * H4file_name</code>	IN: HDF4 filename, the source file
<code>char * H5file_name</code>	IN: HDF5 filename, the target file
<code>int access</code>	IN: The file access flag for the HDF5 file. May contain any one of the following values: <code>H425_CREATE</code> (or 1) Create a new HDF5 file; if the file already exists, the function will fail. <code>H425_OPEN</code> (or 3) Open an existing HDF5 file with read and write permission; if the file does not already exist, create a new HDF5 file. <code>H425_CLOBBER</code> (or 3) If an HDF5 file of the same name already exists, truncate it, i.e., clobber it, and overwrite it with the new file; if the file does not already exist, create a new file. When <code>access</code> contains any value other than the above, <code>H425_CLOBBER</code> is assumed.

**Returns:**

Returns an `h4toh5` identifier if successful; otherwise returns a negative value.

**Non-C API(s):**

**Name:** H4toH5open\_id

**Signature:**

*hid\_t* H4toH5open\_id (*char* \* H4file\_name, *hid\_t* file5\_id)

**Purpose:**

Initializes the H4toH5 library interface.

**Description:**

H4toH5open\_id initializes the H4toH5 library to convert data from one HDF4 file to one HDF5 file. Either this function or H4toH5open must be the first H4toH5 library function called in an application.

In contrast to H4toH5open, a user must first create or open an HDF5 file and optionally specify an HDF5 file property list for that file before passing an HDF5 file identifier to this function.

The h4toh5 identifier returned by this function must be released with H4toH5close or resource leaks will develop.

**Parameters:**

*char* \* H4file\_name      IN: HDF4 filename  
*hid\_t* file5\_id          IN: HDF5 file identifier

**Returns:**

Returns an h4toh5 identifier if successful; otherwise returns a negative value.

**Non-C API(s):**

**Name:** H4toH5close

**Signature:**

*int* H4toH5close (*hid\_t* h5toh5id)

**Purpose:**

Terminates access to the H4toH5 library.

**Description:**

H4toH5close closes all HDF4 and HDF5 object interfaces internally opened by the H4toH5 library and terminates access to the H4toH5 library.

A user application should close all HDF4 and HDF5 object interfaces opened by that application.

**Parameters:**

*hid\_t* h4toh5id IN: h4toh5 identifier

**Returns:**

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

**Non-C API(s):**





## Global attributes

---

**Name:** H4toH5glo\_sds\_attr

**Signature:**

*int* H4toH5glo\_sds\_attr(*hid\_t* h4toh5id)

**Purpose:**

Converts all HDF4 SD interface attributes to attributes of the HDF5 root group.

**Description:**

H4toH5glo\_sds\_attr converts all global SD interface attributes to attributes of the HDF5 root group. The HDF5 attribute names will consist of the prefix GLOSDS\_ appended to the HDF4 attribute names.

**Parameters:**

*hid\_t* h4toh5id      IN: h4toh5 identifier

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5glo\_image\_attr

**Signature:**

*int* H4toH5glo\_image\_attr(*hid\_t* h4toh5id)

**Purpose:**

Converts all HDF4 GR interface attributes to attributes of the HDF5 root group.

**Description:**

H4toH5glo\_image\_attr converts all GR interface attributes to attributes of the HDF5 root group. The HDF5 attribute names consist of the prefix `GLOIMAGE_` appended to the HDF4 attribute names.

**Parameters:**

*hid\_t* h4toh5id      IN: h4toh5 identifier

**Returns:**

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

**Non-C API(s):**

---

## Datatypes

---

**Name:** H4toH5datatype

**Signature:**

```
int H4toH5datatype(hid_t h4toh5id, const int32 h4type, hid_t * h5type_ptr, hid_t *
h5mem_type_ptr, size_t * h4size_ptr, size_t * h4mem_size_ptr)
```

**Purpose:**

Retrieves HDF5 atomic file and memory datatypes and data sizes converted from HDF4 objects.

**Description:**

H4toH5datatype enables the user to retrieve datatype conversion information.

File and memory datatypes, and the size of file and memory datatypes may be represented differently from machine to machine and from memory to disk. H4toH5\_datatypeconv may help users to understand the principals and mechanics of datatype and data size conversion, particularly when doing the conversions on different platforms.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>const int32</i> h4type	IN: HDF4 datatype
<i>hid_t</i> * h5type_ptr	OUT: The address where the converted HDF5 datatype is stored on disk
<i>hid_t</i> * h5memtype_ptr	OUT: The address where the converted HDF5 datatype is stored in memory.
<i>size_t</i> * h4size_ptr	OUT: The address where the converted HDF4 datatype size is stored on disk
<i>size_t</i> * h4mem_size_ptr	OUT: The address where the converted HDF4 datatype size is stored in memory.

**Returns:**

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

**Non-C API(s):**



---

## Dimension scales

---

**Name:** H4toH5one\_dimscale

**Signature:**

```
int H4toH5one_dimscale(hid_t h4toh5id, int32 sds_id, char * h5group_full_path, char *
h5dset_name, char * h5dim_group_full_path, char * h5dim_name, int attr_flag, int ref_flag, int
dim_index)
```

**Purpose:**

Converts an HDF4 dimension scale dataset to an HDF5 dataset.

**Description:**

H4toH5one\_dimscale converts one dimension scale dataset of one SDS object to an HDF5 dataset.

This function also provides a mechanism for associating the dimension scale dataset with a converted SDS or other equivalent HDF5 datasets. The association is made through an HDF5 dataset attribute that stores an object reference from the HDF5 dataset to dimension scale dataset. (See *Mapping HDF4 Objects to HDF5 Objects* [1], section 3.1 and figure 1, for details.)

The `ref_flag` parameter controls whether the HDF5 dataset converted from the dimension scale is associated with the HDF5 dataset converted from the SDS or with other HDF5 datasets. The user can choose to convert only the dimension scale dataset by setting the `ref_flag` parameter to `H425_NOREF`. If the user wants the converted dimension scale to be associated with another HDF5 dataset, the HDF5 dataset name must be provided in `h5dset_name`.

The `attr_flag` parameter specifies whether attributes of the dimension scale are to be converted.

Note: HDF5 dimension scale specification is currently under development. This API is subject to change to maintain consistency once the HDF5 dimension scale specification is finalized.

**Parameters:**

<code>hid_t</code> h4toh5id	IN: h4toh5 identifier
<code>int32</code> sds_id	IN: SDS identifier with which the SDS dimension scale dataset is associated in the HDF4 file
<code>char</code> * h5group_full_path	IN: The absolute pathname of the HDF5 group of which one member will generate an object reference to the converted dimension scale dataset. Must start with "/".
<code>char</code> * h5dset_name	IN: The relative pathname of the HDF5 dataset containing the converted SDS or another equivalent HDF5 dataset. An object reference to the converted dimension scale dataset will be generated for this HDF5 dataset. If <code>h5dset_name</code> is set to NULL, no object reference attribute will be generated to associate the converted dimension scale with any HDF5 dataset.

<i>char</i> * h5dim_group_full_path	IN: The absolute pathname of the HDF5 group to which the converted dimension scale dataset will be assigned. Must start with "/" unless set to NULL. If set to NULL, the default dimension group name /HDF4_DIMGROUP will be used.
<i>char</i> * h5dim_name	IN: The relative pathname of the HDF5 dataset to which the dimension scale dataset will be converted. h5dim_name is relative to h5dim_group_full_path. If set to NULL, a default dimension scale dataset name will be generated based on the dimension scale dataset name in the HDF4 file.
<i>int</i> ref_flag	IN: A flag indicating whether the dimension scale dataset is associated with the HDF5 dataset specified in h5dset_name. H425_NOREF (or 0) No object reference is generated. H425_REF (or 1) The object reference from the dataset specified in h5dset_name to this dimension scale dataset is generated.
<i>int</i> attr_flag	IN: A flag specifying whether attributes of the dimension scale dataset are to be converted. H425_NOATTRS (or 0) No attributes are converted. H425_ALLATTRS (or 1) All attributes are converted to attributes of the corresponding HDF5 objects.
<i>int</i> dim_index	IN: An index indicating which dimension scale dataset will be converted. Must be a value from 0 (zero), for the first dimension scale, to n-1, for the last dimension scale, where n is the number of SDS dimensions.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5all\_dimscale

**Signature:**

```
int H4toH5all_dimscale(hid_t h4toh5id, int32 sds_id, char * h5group_full_path, char *
h5dset_name, char * h5dim_group_full_path, int attr_flag, int ref_flag)
```

**Purpose:**

Converts all HDF4 dimension scale datasets to HDF5 datasets.

**Description:**

H4toH5all\_dimscale converts all dimension scale datasets associated with one SDS object to HDF5 datasets.

This function also provides a mechanism to associate the dimension scale datasets with the converted SDS object or another equivalent HDF5 dataset. The association is made through an attribute of the HDF5 dataset that stores an object reference from the HDF5 dataset to the dimension scale dataset. (See *Mapping HDF4 Objects to HDF5 Objects* [1], section 3.1 and figure 1, for details.)

The `ref_flag` parameter specifies whether the HDF5 datasets containing the converted dimension scales are to be associated with another HDF5 dataset. If the dimension scales are to be converted only (i.e., and not associated with another HDF5 dataset), `ref_flag` is set to `H425_NOREF`. If the converted dimension scales are to be associated with another HDF5 dataset, `ref_flag` is set to `H425_REF` and the dataset is specified in `h5dset_name`.

The `attr_flag` parameter specifies whether attributes of the dimension scale are to be converted.

Note: HDF5 dimension scale specification is currently under development. This API is subject to change to maintain consistency once the HDF5 dimension scale specification is finalized.

**Parameters:**

<code>hid_t</code> h4toh5id	IN: h4toh5 identifier
<code>int32</code> sds_id	IN: SDS identifier with which the converted SDS dimension scale datasets are associated in the HDF4 file
<code>char</code> * h5group_full_path	IN: The absolute pathname of the target HDF5 group. Object references to the converted dimension scale datasets will be generated for the member of this group specified in <code>h5dset_name</code> . Must start with <code>"/"</code> .
<code>char</code> * h5dset_name	IN: The relative pathname of the target HDF5 dataset, relative to <code>h5group_full_path</code> . Object references to the converted dimension scale datasets will be generated for this dataset. If <code>h5dset_name</code> is set to <code>NULL</code> , no object reference attributes will be generated.
<code>char</code> * h5dim_group_full_path	IN: The absolute pathname of the HDF5 group to which the converted dimension scale datasets are to be assigned. Must start with <code>"/"</code> . If set to <code>NULL</code> , the default dimension scale group name <code>/HDF4_DIMGROUP</code> will be used.

*int* ref\_flag

IN: A flag specifying whether the converted dimension scale datasets are to be associated with another HDF5 dataset:

H425\_NOREF (or 0)

No

H425\_REF (or 1)

Yes

H425\_NOREF (or 0)

No object reference is generated.

H425\_REF (or 1)

The object reference from the dataset specified in `h5dset_name` to this dimension scale dataset is generated.

*int* attr\_flag

IN: A flag specifying whether dimension scale attributes are to be converted:

H425\_NOATTRS (or 0)

No attributes are converted.

H425\_ALLATTRS (or 1)

All attributes are converted to attributes of the corresponding HDF5 objects.

**Returns:**

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

**Non-C API(s):**



---

## Annotations

---

**Name:** H4toH5anno\_file\_label

**Signature:**

*int* H4toH5anno\_file\_label(*hid\_t* h4toh5id, *char* \* anno\_label\_name, *int* label\_index)

**Purpose:**

Converts an HDF4 file label to an HDF5 root group attribute

**Description:**

H4toH5anno\_file\_label converts an HDF4 file label annotation to an attribute of the root group of the target HDF5 file.

If anno\_label\_name is set to NULL, a default attribute name is created as the string HDF4\_FILE\_LABEL\_index, where index is the index of the file label.

If label\_index is beyond the range of valid HDF4 file label annotations, a warning message will be generated and nothing will be converted.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>char</i> * anno_label_name	IN: The attribute name in the HDF5 file of the converted file label.
<i>int</i> label_index	IN: The HDF4 index of the label that is to be converted. Must be a value from 0 (zero), for the first file label, to n-1, for the last file label, where n is the number of file labels.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5anno\_file\_desc

**Signature:**

*int* H4toH5anno\_file\_desc(*hid\_t* h4toh5id, *char* \* anno\_desc\_name, *int* desc\_index)

**Purpose:**

Converts an HDF4 file description to an attribute of the HDF5 root group.

**Description:**

H4toH5annofil\_desc converts an HDF4 file description annotation to an attribute of the root group of the target HDF5 file.

If the anno\_desc\_name parameter is set to NULL, a default attribute name is created as the string HDF4\_FILE\_DESCRIPTION\_index, where index is the HDF4 index of the file description annotation.

If desc\_index is beyond the range of valid HDF4 file description annotations, a warning message will be generated and nothing will be converted.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>char</i> * anno_desc_name	IN: The attribute name in the HDF5 file of the converted file description annotation
<i>int</i> desc_index	IN: The HDF4 index of the file description annotation that is to be converted. Must be a value from 0 (zero), for the first file description, to n-1, for the last file description, where n is the number of file descriptions.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5anno\_file\_all\_labels

**Signature:**

*int* H4toH5anno\_file\_all\_labels(*hid\_t* h4toh5id)

**Purpose:**

Converts all HDF4 file labels to HDF5 root group attributes.

**Description:**

H4toH5anno\_file\_all\_labels converts all HDF4 file label annotations to attributes of the root group of the target HDF5 file.

The name of each attribute is created as the string `HDF4_FILE_LABEL_index`, where *index* is the HDF4 index of the file label annotation.

**Parameters:**

*hid\_t* h4toh5id      IN: h4toh5 identifier

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5anno\_file\_all\_descs

**Signature:**

*int* H4toH5anno\_file\_all\_descs(*hid\_t* h4toh5id)

**Purpose:**

Converts all HDF4 file descriptions to HDF5 root group attributes.

**Description:**

H4toH5anno\_file\_all\_descs converts all HDF4 file description annotations to attributes of the root group of the target HDF5 file.

The name of each attribute is created as the string HDF4\_FILE\_DESCRIPTION\_*index*, where *index* is the index of the file description annotation.

**Parameters:**

*hid\_t* h4toh5id      IN: h4toh5 identifier

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5anno\_obj\_label

**Signature:**

```
int H4toH5anno_obj_label(hid_t h4toh5id, char * h5group_full_path, char * h5dset_name,
uint16 obj_ref, int32 obj_tag, char * anno_label_name, int label_index)
```

**Purpose:**

Converts an HDF4 object label to an HDF5 object attribute.

**Description:**

H4toH5anno\_obj\_label converts an HDF4 object label annotation to an attribute of the corresponding HDF5 dataset or group.

H4toH5anno\_obj\_label passes the absolute pathname of an HDF5 group in `h5group_full_path`. If the object label to be converted is a group annotation, `h5dset_name` is set to `NULL` and the converted label will be associated with the group indicated by `h5group_full_path`. If the object label to be converted is a dataset annotation, the converted label will be associated with the dataset specified in `h5dset_name`.

If `anno_label_name` is set to `NULL`, a default attribute name is created as the string `HDF4_OBJECT_LABEL_index`, where `index` is the index of the object label.

Only object label annotations associated with the following basic HDF4 objects can be converted: Vgroups, Vdatas, SDSs, images, and palettes.

If `label_index` is beyond the range of valid HDF4 object label annotations, a warning message will be generated and nothing will be converted.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>char</i> * h5group_full_path	IN: The absolute pathname of the target HDF5 group. The converted object attribute will be associated with either this group or the member of this group specified in <code>h5dset_name</code> .
<i>char</i> * h5dset_name	IN: The relative pathname of the target HDF5 dataset, relative to <code>h5group_full_path</code> . The object attribute will be associated with this dataset; if the object attribute is to be associated with the group, <code>h5dset_name</code> must be set to <code>NULL</code> .
<i>uint16</i> obj_ref	IN: The HDF4 reference number of the HDF4 object with which the object label annotation is associated.
<i>int32</i> obj_tag	IN: The HDF4 tag of the HDF4 object with which the object label annotation is associated.
<i>char</i> * anno_label_name	IN: The name of the HDF5 attribute containing the converted object label.
<i>int</i> label_index	IN: The HDF4 index of the object label that is to be converted. Must be a value from 0 (zero), for the first object label, to $n-1$ , for the last object label, where $n$ is the number of object label annotations.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5anno\_obj\_desc

**Signature:**

```
int H4toH5anno_obj_desc(hid_t h4toh5id, char * h5group_full_path, char * h5dset_name, uint16
bj_ref, int32 obj_tag, char * anno_desc_name, int desc_index)
```

**Purpose:**

Converts an HDF4 object description to HDF5 object attribute.

**Description:**

H4toH5anno\_obj\_desc converts an HDF4 object description annotation to an attribute of the corresponding HDF5 dataset or group.

This function passes the absolute pathname of an HDF5 group in `h5group_full_path`. If the object description to be converted is a group annotation, `h5dset_name` is set to `NULL` and the converted description will be associated with the group indicated by `h5group_full_path`. If the object description to be converted is a dataset annotation, the converted description will be associated with the dataset specified in `h5dset_name`.

If the `anno_desc_name` parameter is not provided, a default attribute name is created as the string `HDF4_OBJECT_DESCRIPTION_index`, where `index` is the index of the object description.

Only object description annotations associated with the following basic HDF4 objects can be converted: Vgroups, Vdatas, SDSs, images, and palettes.

If `desc_index` is beyond the range of valid HDF4 object description annotations, a warning message will be generated and nothing will be converted.

**Parameters:**

<code>hid_t h4toh5id</code>	IN: h4toh5 identifier
<code>char * h5group_full_path</code>	IN: The absolute pathname of the target HDF5 group. The object description will be associated with either this group or the member of this group specified in <code>h5dset_name</code> .
<code>char * h5dset_name</code>	IN: The relative pathname of the target HDF5 dataset, relative to <code>h5group_full_path</code> . The object attribute will be associated with this dataset; if the object attribute is to be associated with the group, <code>h5dset_name</code> must be set to <code>NULL</code> .
<code>uint16 obj_ref</code>	IN: The HDF4 reference number of the HDF4 object with which the object description annotation is associated.
<code>int32 obj_tag</code>	IN: The HDF4 tag of the HDF4 object with which the object description annotation is associated.
<code>char * anno_descname</code>	IN: The name of the HDF5 attribute containing the converted object description.
<code>int desc_index</code>	IN: The HDF4 index of the object description that is to be converted. Must be a value from 0 (zero), for the first object description, to $n-1$ , for the last object description, where $n$ is the number of object description annotations.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5anno\_obj\_all\_labels

**Signature:**

```
int H4toH5anno_obj_all_labels(hid_t h4toh5id, char * h5group_full_path, char * h5dset_name,
uint16 obj_ref, int32 obj_tag)
```

**Purpose:**

Converts all label annotations of an HDF4 object to HDF5 attributes.

**Description:**

H4toH5anno\_obj\_all\_labels converts all object label annotations associated with an HDF4 object to attributes of the corresponding HDF5 dataset or group.

H4toH5anno\_obj\_all\_labels passes the absolute pathname of an HDF5 group in h5group\_full\_path. If the object labels to be converted are group annotations, h5dset\_name is set to NULL and the converted labels will be associated with the group indicated by h5group\_full\_path. If the object labels to be converted are dataset annotations, the converted labels will be associated with the dataset specified in h5dset\_name.

Attribute names are created as the string HDF4\_OBJECT\_LABEL\_index, where index is the index of the object label.

Only object label annotations associated with the following basic HDF4 objects can be converted: Vgroups, Vdatas, SDSs, images, and palettes.

**Parameters:**

hid_t h4toh5id	IN: h4toh5 identifier
char * h5group_full_path	IN: The absolute pathname of the target HDF5 group. The object label attributes will be associated with either this group or the member of this group specified in h5dset_name.
char * h5dset_name	IN: The relative pathname of the target HDF5 dataset, relative to h5group_full_path. The object attributes will be associated with this dataset; if the object attributes are to be associated with the group, h5dset_name must be set to NULL.
uint16 obj_ref	IN: The HDF4 reference number of the HDF4 object with which the object label annotations are associated.
int32 obj_tag	IN: The HDF4 tag of the HDF4 object with which the object label annotations are associated.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5anno\_obj\_all\_descs

**Signature:**

*int* H4toH5anno\_obj\_all\_descs(*hid\_t* h4toh5id, *char* \* h5group\_full\_path, *char* \* h5dset\_name, *uint16* obj\_ref, *int32* obj\_tag)

**Purpose:**

Converts all description annotations of an HDF4 object to HDF5 attributes.

**Description:**

H4toH5anno\_obj\_all\_descs converts all description annotations associated with an HDF4 object to attributes of the corresponding HDF5 dataset or group.

H4toH5anno\_obj\_all\_descs passes the absolute pathname of an HDF5 group in h5group\_full\_path. If the object descriptions to be converted are group annotations, h5dset\_name is set to NULL and the converted labels will be associated with the group indicated by h5group\_full\_path. If the object descriptions to be converted are dataset annotations, the converted labels will be associated with the dataset specified in h5dset\_name.

Default attribute names are created as the string HDF4\_OBJECT\_DESCRIPTION\_index, where index is the object description index.

Only object description annotations associated with the following basic HDF4 objects can be converted: Vgroups, Vdatas, SDSs, images, and palettes.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>char</i> * h5group_full_path	IN: The absolute pathname of the target HDF5 group. The object description attributes are associated with either this group or the member of this group specified in h5dset_name.
<i>char</i> * h5dset_name	IN: The relative pathname of the target HDF5 dataset, relative to h5group_full_path. The object attributes will be associated with this dataset; if the object attributes are to be associated with the group, h5dset_name must be set to NULL.
<i>uint16</i> obj_ref	IN: The HDF4 reference number of the HDF4 object with which the object description annotations are associated.
<i>int32</i> obj_tag	IN: The HDF4 tag of the HDF4 object with which the object description annotations are associated.

**Returns:**

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

**Non-C API(s):**



---

## Error messages

---

**Name:** H4toH5error\_get

**Signature:**

*int* H4toH5error\_get(*hid\_t* h4toh5id)

**Purpose:**

Prints error information.

**Description:**

H4toH5error\_get prints error information to standard output.

The error information includes the following:

- ◇ Error level number
- ◇ Error message - a brief message describing the error
- ◇ Error code number - see the *H4toH5 Conversion Library User's Guide* for error code descriptions
- ◇ Line at which the error occurred
- ◇ File in which the error occurred

This function is provided primarily for debugging purposes.

**Parameters:**

*hid\_t* h4toh5id      IN: h4toh5 identifier

**Returns:**

Should always return 0 (zero).

**Non-C API(s):**



---

## Vgroups

---

**Name:** H4toH5bas\_vgroup

**Signature:**

```
int H4toH5bas_vgroup (hid_t h4toh5id, int32 vgroup_id, char * h5parent_group_full_path,
char * h5child_group_name, int vg_flag, int attr_flag)
```

**Purpose:**

Converts an HDF4 Vgroup to an HDF5 group.

**Description:**

H4toH5bas\_vgroup converts an HDF4 Vgroup to an HDF5 group.

Depending on the setting of the `vg_flag` parameter, either the Vgroup only or the Vgroup and its non-Vgroup members will be converted to corresponding HDF5 objects.

H4toH5bas\_vgroup is designed for use with a single Vgroup. Use H4toH5adv\_group to recursively convert a Vgroup and member Vgroups to HDF5 objects.

**Parameters:**

<code>hid_t h4toh5id</code>	IN: h4toh5 identifier
<code>int32 vgroup_id</code>	IN: Vgroup identifier in the HDF4 file
<code>char * h5parent_group_full_path</code>	IN: The absolute pathname of the HDF5 group of which the converted Vgroup is to be a member. Must start with "/".
<code>char * h5child_group_name</code>	IN: The relative pathname of the converted Vgroup, relative to <code>h5parent_group_full_path</code> . If set to NULL, the H4toH5 library generates a default HDF5 group name based on the HDF4 Vgroup name.
<code>int vg_flag</code>	IN: a flag specifying whether non-Vgroup members of the vgroup are to be converted: H425_NONMEMBERS (or 0) Non-Vgroup members are not converted. H425_ALLMEMBERS (or 1) All non-Vgroup members of this Vgroup are converted to corresponding HDF5 objects.
<code>int attr_flag</code>	IN: A flag specifying whether attributes of the converted vgroup and its non-Vgroup members are to be converted: H425_NOATTRS (or 0) No attributes are converted. H425_ALLATTRS (or 1) All attributes are converted to attributes of the corresponding HDF5 objects.

**Returns:**

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

**Name:** H4toH5vgroup\_attr\_name

**Signature:**

```
int H4toH5vgroup_attr_name (hid_t h4toh5id, int32 vgroup_id, char *
h5parent_group_full_path, char * h5child_group_name, char * attr_name)
```

**Purpose:**

Converts an HDF4 Vgroup attribute to an HDF5 group attribute.

**Description:**

H4toH5vgroup\_attr\_name converts an HDF4 Vgroup attribute, specified by its HDF4 name, to an HDF5 group attribute.

This function assumes the preexistence of an HDF5 group that the converted Vgroup attribute will belong to. If h5child\_group\_name is set to NULL, the HDF4 Vgroup specified in vgroup\_id should be converted before this function is called.

The attr\_name parameter specifies which Vgroup attribute is to be converted.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>int32</i> vgroup_id	IN: HDF4 Vgroup identifier
<i>char *</i> h5parent_group_full_path	IN: The absolute pathname of an HDF5. Must start with "/". The converted Vgroup attribute will be associated with a member of this group as specified in h5child_group_name.
<i>char *</i> h5child_group_name	IN: The relative pathname of the target HDF5 group, relative to h5parent_group_full_path. If set to NULL, the H4toH5 library will generate a default HDF5 group name based on the HDF4 Vgroup name.
<i>char *</i> attr_name	IN: The HDF4 name of the Vgroup attribute to be converted.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5vgroup\_attr\_index

**Signature:**

```
int H4toH5vgroup_attr_index (hid_t h4toh5id, int32 vgroup_id, char *
h5parent_group_full_path, char * h5child_group_name, int attr_index)
```

**Purpose:**

Converts an HDF4 Vgroup attribute to an HDF5 group attribute.

**Description:**

H4toH5vgroup\_attr\_index converts an HDF4 Vgroup attribute, specified by its HDF4 index, to an HDF5 group attribute.

This function assumes the preexistence of an HDF5 group that the converted Vgroup attribute will belong to. If h5child\_group\_name is set to NULL, The HDF4 Vgroup specified in vgroup\_id should be converted before this function is called.

The attr\_index parameter specifies which Vgroup attribute to be converted.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>int32</i> vgroup_id	IN: HDF4 Vgroup identifier
<i>char</i> * h5parent_group_full_path	IN: The absolute pathname of an HDF5 group. Must start with "/". The converted Vgroup attribute will be associated with a member of this group as specified in h5child_group_name.
<i>char</i> * h5child_group_name	IN: The relative pathname of the target HDF5 group, relative to h5parent_group_full_path. If set to NULL, the H4toH5 library will generate a default HDF5 group name based on the name of the HDF4 Vgroup object.
<i>int</i> attr_index	IN: The HDF4 index of the Vgroup attribute to be converted. Must be a value from 0 (zero), for the first Vgroup attribute, to n-1, for the last Vgroup attribute, where n is the number of HDF4 Vgroup attributes.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5adv\_group

**Signature:**

```
int H4toH5adv_group (hid_t h4toh5id, int32 vgroup_id, char * h5parent_group_full_path, char *
h5child_group_name)
```

**Purpose:**

Converts an HDF4 Vgroup and its descendents to an HDF5 group.

**Description:**

H4toH5adv\_group converts an HDF4 Vgroup and all its descendents to an HDF5 group and its members.

Users should be extremely careful when using this function, especially when the Vgroup structure to be converted is complicated.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>int32</i> vgroup_id	IN: Vgroup identifier in the HDF4 file
<i>char</i> * h5parent_group_full_path	IN: The absolute pathname of the group of which the converted Vgroup is to be a member. Must start with "/".
<i>char</i> * h5child_group_name	IN: The relative pathname of the converted Vgroup, relative to h5parent_group_full_path. If set to NULL, the H4toH5 library generates a default HDF5 group name based on HDF4 Vgroup name.

**Returns:**

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

**Non-C API(s):**

---

## Vdatas

---

**Name:** H4toH5vdata

**Signature:**

```
int H4toH5vdata(hid_t h4toh5id, int32 vdata_id, char * h5group_full_path, char * h5dset_name,
               int attr_flag)
```

**Purpose:**

Converts an HDF4 Vdata object to an HDF5 dataset.

**Description:**

H4toH5vdata converts an independent Vdata object to an HDF5 dataset. The `attr_flag` parameter controls the conversion of Vdata attributes.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>int32</i> vdata_id	IN: Vdata identifier in the HDF4 file
<i>char</i> * h5group_full_path	IN: The absolute pathname of the HDF5 group to which the converted dataset will be assigned. Must start with "/".
<i>char</i> * h5dset_name	IN: The relative pathname of the target HDF5 dataset, relative to <code>h5group_full_path</code> . The Vdata is converted to this dataset. If <code>h5dset_name</code> is set to NULL, the H4toH5 library will generate a default HDF5 dataset name based on the Vdata name.
<i>int</i> attr_flag	IN: a flag specifying whether Vdata attributes are to be converted: H425_NOATTRS (or 0) No attributes are converted. H425_ALLATTRS (or 1) All attributes are converted to attributes of the corresponding HDF5 dataset.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5vdata\_attr\_name

**Signature:**

```
int H4toH5vdata_attr_name(hid_t h4toh5id, int32 vdata_id, char * h5group_full_path, char *
h5dset_name, char* attr_name)
```

**Purpose:**

Converts an HDF4 Vdata object attribute to an HDF5 dataset attribute.

**Description:**

H4toH5vdata\_attr\_name converts an HDF4 Vdata attribute, specified by its HDF4 name, to an HDF5 dataset attribute.

This function assumes the preexistence of an HDF5 dataset that the converted Vdata attribute will belong to. If h5dset\_name is set to NULL, the HDF4 Vdata specified in vdata\_id should be converted before this function is called.

The attr\_name parameter specifies which SDS attribute is to be converted.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>int32</i> vdata_id	IN: HDF4 Vdata identifier
<i>char</i> * h5group_full_path	IN: The absolute pathname of an HDF5 group. Must start with "/". The converted Vdata attribute will be associated with a member of this group as specified in h5dset_name.
<i>char</i> * h5dset_name	IN: The relative pathname of an HDF5 dataset, relative to h5group_full_path. If set to NULL, the H4toH5 library will generate a default HDF5 dataset name based on the HDF4 Vdata name.
<i>char</i> * attr_name	IN: The HDF4 name of the Vdata attribute to be converted.

**Returns:**

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

**Non-C API(s):**



**Name:** H4toH5vdata\_attr\_index

**Signature:**

```
int H4toH5vdata_attr_index(hid_t h4toh5id, int32 vdata_id, char * h5group_full_path, char *
h5dset_name, int attr_index)
```

**Purpose:**

Converts an HDF4 Vdata object attribute to an HDF5 dataset attribute.

**Description:**

H4toH5vdata\_attr\_index converts an HDF4 Vdata attribute, specified by its HDF4 index, to an HDF5 dataset attribute.

This function assumes the preexistence of an HDF5 dataset that the converted Vdata attribute will belong to. If h5dset\_name is set to NULL, the HDF4 Vdata specified in vdata\_id should be converted before this function is called.

The attr\_index parameter specifies which Vdata attribute is to be converted.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>int32</i> vdata_id	IN: HDF4 Vdata identifier
<i>char</i> * h5group_full_path	IN: The absolute pathname of an HDF5 group. Must start with "/". The converted Vdata attribute will be associated with a dataset of this group as specified in h5dset_name.
<i>char</i> * h5dset_name	IN: The relative pathname of the target HDF5 dataset, relative to h5group_full_path. If set to NULL, the H4toH5 library will generate a default HDF5 dataset name based on the HDF4 Vdata name.
<i>int</i> attr_index	IN: The HDF4 index of the Vdata attribute to be converted. Must be a value from 0 (zero), for the first Vdata attribute, to n-1, for the last Vdata attribute, where n is the number of HDF4 Vdata attributes.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5vdata\_field\_attr\_name

**Signature:**

```
int H4toH5vdata_field_attr_name(hid_t h4toh5id, int32 vdata_id, char * h5group_full_path,
char * h5dset_name, char * field_name, char * field_attr_name)
```

**Purpose:**

Converts an HDF4 Vdata field attribute to an HDF5 dataset attribute.

**Description:**

H4toH5vdata\_field\_attr\_name converts an object attribute associated with an HDF4 Vdata field to an HDF5 dataset attribute. The Vdata field and the associated attribute are specified by their HDF4 names.

This function assumes the preexistence of an HDF5 dataset that the converted Vdata field attribute will belong to. If h5dset\_name is set to NULL, the HDF4 Vdata specified in vdata\_id should be converted before this function is called.

The parameters field\_name and field\_attrname, respectively, specify the Vdata field name and the associated object attribute name.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>int32</i> vdata_id	IN: HDF4 Vdata identifier
<i>char</i> * h5group_full_path	IN: The absolute pathname of an HDF5 group. Must start with "/". The converted Vdata field attribute will be associated with a member of this group as specified in h5dset_name.
<i>char</i> * h5dset_name	IN: The relative pathname of HDF5 dataset, relative to h5group_full_path. If set to NULL, the H4toH5 library will generate a default HDF5 dataset name based on the HDF4 Vdata name.
<i>char</i> * field_name	IN: The HDF4 name of the Vdata field to be converted.
<i>char</i> * field_attr_name	IN: The HDF4 name of the Vdata field attribute to be converted.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5vdata\_field\_attr\_index

**Signature:**

```
int H4toH5vdata_field_attr_index(hid_t h4toh5id, int32_t vdata_id, char * h5group_full_path,
char * h5dset_name, int field_index, int field_attr_index)
```

**Purpose:**

Converts an HDF4 Vdata field object attribute into an HDF5 dataset attribute.

**Description:**

H4toH5vdata\_field\_attr\_index converts an object attribute associated with an HDF4 Vdata field to an HDF5 dataset attribute. The Vdata field and the associated attribute are specified by their HDF4 indexes.

This function assumes the preexistence of an HDF5 dataset that the converted Vdata field attribute will belong to. If h5dset\_name is set to NULL, the HDF4 Vdata specified in vdata\_id should be converted before this function is called.

The parameters field\_index and field\_attr\_index, respectively, specify the Vdata field index and the associated object attribute index.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>int32_t</i> vdata_id	IN: HDF4 Vdata identifier
<i>char</i> * h5group_full_path	IN: The absolute pathname of an HDF5 group. Must start with "/". The converted Vdata field attribute will be associated with a member of this group as specified in h5dset_name.
<i>char</i> * h5dset_name	IN: The relative pathname of HDF5 dataset, relative to h5group_full_path. If set to NULL, the H4toH5 library will generate a default HDF5 dataset name based on the HDF4 Vdata name.
<i>int</i> field_index	IN: The HDF4 index of the Vdata field to be converted. Must be a value from 0 (zero), for the first Vdata field, to n-1, for the last Vdata field, where n is the number of HDF4 Vdata fields.
<i>int</i> field_attr_index	IN: The HDF4 index of the Vdata field attribute to be converted. Must be a value from 0 (zero), for the first Vdata field attribute, to n-1, for the last Vdata field attribute, where n is the number of HDF4 Vdata field attributes.

**Returns:**

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

**Non-C API(s):**



---

## SDSs

---

**Name:** H4toH5sds

**Signature:**

```
int H4toH5sds(hid_t h4toh5id, int32 sds_id, char * h5group_full_path, char * h5dset_name,
char * h5dim_group_full_path, int dim_flag, int attr_flag)
```

**Purpose:**

Converts an HDF4 SDS object into an HDF5 dataset.

**Description:**

H4toH5sds converts an SDS object in the HDF4 file to an HDF5 dataset in the HDF5 file.

Depending on the settings of the `dim_flag` and `attr_flag` parameters, the dimension scale datasets and attributes of the SDS object may or may not be converted.

If dimension scale datasets are converted, the `attr_flag` parameter controls the conversion of their attributes.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>int32</i> sds_id	IN: SDS identifier in the HDF4 file
<i>char</i> * h5group_full_path	IN: The absolute pathname of the HDF5 group into which the converted SDS is to be placed. Must start with "/".
<i>char</i> * h5dset_name	IN: The relative pathname of the HDF5 dataset, relative to <code>h5group_full_path</code> . If set to NULL, the H4toH5 library generates a default HDF5 dataset name based on the SDS name.
<i>char</i> * h5dim_group_full_path	IN: The absolute pathname of the HDF5 group into which the converted dimension scale datasets are to be placed. If set to NULL, the default dimension group name <code>/HDF4_DIMGROUP</code> is used.
<i>int</i> dim_flag	IN: A flag specifying whether dimension scale datasets are to be converted: H425_NODIMSCALE (or 0) The dimension scale datasets are not converted. H425_DIMSCALE (or 1) The dimension scale datasets are converted to corresponding HDF5 objects.
<i>int</i> attr_flag	IN: A flag specifying whether SDS attributes are to be converted H425_NOATTRS (or 0) No attributes are converted. H425_ALLATTRS (or 1) All attributes are converted to attributes of the corresponding HDF5 objects; if <code>dim_flag</code> is set, attributes of the dimension scale datasets are also converted.

**Returns:**

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

**Name:** H4toH5sds\_attr\_name

**Signature:**

```
int H4toH5sds_attr_name(hid_t h4toh5id, int32 sds_id, char * h5group_full_path, char *
h5dset_name, char * attr_name)
```

**Purpose:**

Converts an HDF4 SDS object attribute to an HDF5 dataset attribute.

**Description:**

H4toH5sds\_attr\_name converts an HDF4 SDS attribute, specified by its HDF4 name, to an HDF5 dataset attribute.

This function assumes the preexistence of an HDF5 dataset that the converted SDS attribute will belong to. If h5dset\_name is set to NULL, the HDF4 SDS specified in sds\_id should be converted before this function is called.

The attr\_name parameter specifies which SDS attribute is to be converted.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>int32</i> sds_id	IN: HDF4 SDS identifier
<i>char</i> * h5group_full_path	IN: The absolute pathname of an HDF5 group. Must start with "/". The converted SDS attribute will be associated with a dataset of this group as specified in h5dset_name.
<i>char</i> * h5dset_name	IN: The relative pathname of target HDF5 dataset, relative to h5group_full_path. If set to NULL, the H4toH5 library will generate a default HDF5 dataset name based on the HDF4 SDS name.
<i>char</i> * attr_name	IN: The name of the HDF4 SDS attribute to be converted.

**Returns:**

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

**Name:** H4toH5sds\_attr\_index

**Signature:**

```
int H4toH5sds_attr_index(hid_t h4toh5id, int32 sds_id, char * h5group_full_path, char *
h5dset_name, int attr_index)
```

**Purpose:**

Converts an HDF4 SDS object attribute to an HDF5 dataset attribute.

**Description:**

H4toH5sds\_attr\_index converts an HDF4 SDS attribute, specified by its HDF4 index, to an HDF5 dataset attribute.

This function assumes the preexistence of an HDF5 dataset that the converted SDS attribute will belong to. If `h5dset_name` is set to NULL, the HDF4 SDS specified in `sds_id` should be converted before this function is called.

The `attr_index` parameter specifies which SDS attribute is to be converted.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>int32</i> sds_id	IN: HDF4 SDS identifier
<i>char</i> * h5group_full_path	IN: The absolute pathname of HDF5 group. Must start with "/". The converted SDS attribute will be associated with a member of this group as specified in <code>h5dset_name</code> .
<i>char</i> * h5dset_name	IN: The relative pathname of the target HDF5 dataset, relative to <code>h5group_full_path</code> . If set to NULL, the H4toH5 library will generate a default HDF5 dataset name based on the HDF4 SDS name.
<i>int</i> attr_index	IN: The HDF4 index of the SDS attribute to be converted. Must be a value from 0 (zero), for the first SDS attribute, to $n-1$ , for the last SDS attribute, where $n$ is the number of HDF4 SDS attributes.

**Returns:**

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

**Non-C API(s):**





---

## Images and palettes

---

**Name:** H4toH5image

**Signature:**

```
int H4toH5image(hid_t h4toh5id, int32 ri_id, char * h5group_full_path, char * h5dset_name,
char * h5pal_group_full_path, char * h5pal_name, int attr_flag, int pal_flag)
```

**Purpose:**

Converts an HDF4 image object to an HDF5 dataset.

**Description:**

H4toH5image converts an HDF4 image object to an HDF5 dataset. Depending on the settings of the `attr_flag` and `pal_flag` parameters, the associated palette and attributes may be converted. If the palette is converted, the `attr_flag` parameter also controls the conversion of the associated palette attributes.

**Parameters:**

<code>hid_t h4toh5id</code>	IN: h4toh5 identifier
<code>int32 ri_id</code>	IN: Raster image identifier in the HDF4 file
<code>char * h5group_full_path</code>	IN: The absolute pathname of the HDF5 group to which the converted image dataset will be assigned. Must start with "/".
<code>char * h5dset_name</code>	IN: The relative pathname of an HDF5 dataset, relative to <code>h5group_full_path</code> . If <code>h5dset_name</code> is set to NULL, the H4toH5 library will generate a default HDF5 dataset name based on the name of the converted image.
<code>char * h5pal_group_full_path</code>	IN: The absolute pathname of the HDF5 group to which the converted palette will be assigned. If the pathname is not set to NULL, it must start with "/". If set to NULL, a default palette group name <code>/HDF4_PALROUP</code> will be used.
<code>char * h5pal_name</code>	IN: The relative pathname of the target HDF5 dataset, relative to <code>h5pal_group_full_path</code> . The palette will be converted to this dataset. If <code>h5pal_name</code> is set to NULL, a default palette dataset name is created as the string <code>HDF4_PALETTE_ref</code> , where <code>ref</code> is the reference number of the palette object in the HDF4 file.
<code>int pal_flag</code>	IN: A flag specifying whether a palette associated with the image is to be converted: H425_NOPAL (or 0) The palette is not converted. H425_PAL (or 1) The palette is converted.

*int* attr\_flag

IN: A flag to determine whether image attributes are to be converted:

H425\_NOATTRS (or 0)

No attributes are converted.

H425\_ALLATTRS (or 1)

All attributes are converted. If `pal_flag` parameter is set to H425\_PAL, palette attributes are also converted.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5image\_attr\_name

**Signature:**

```
int H4toH5image_attr_name(hid_t h4toh5id, int32 ri_id, char * h5group_full_path, char *
h5dset_name, char * attr_name)
```

**Purpose:**

Converts an HDF4 image object attribute to an HDF5 dataset attribute.

**Description:**

H4toH5image\_attr\_name converts an HDF4 image attribute, specified by its HDF4 name, to an HDF5 dataset attribute.

This function assumes the preexistence of an HDF5 dataset that the converted image attribute will belong to. If h5dset\_name is set to NULL, the HDF4 image specified in ri\_id should be converted before this function is called.

The attr\_name parameter specifies which image attribute is to be converted.

This function works for the following types of HDF4 images: GR, RI8, and RI24.

**Parameters:**

hid_t h4toh5id	IN: h4toh5 identifier
int32 ri_id	IN: HDF4 raster image identifier
char * h5group_full_path	IN: The absolute pathname of an HDF5 group. Must start with "/". The converted image attribute will be associated with a dataset of this group as specified in h5dset_name.
char * h5dset_name	IN: The relative pathname of an HDF5 dataset, relative to h5group_full_path. If set to NULL, the H4toH5 library will generate a default HDF5 dataset name based on the HDF4 image name.
char * attr_name	IN: The HDF4 name of the image attribute to be converted.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5image\_attr\_index

**Signature:**

```
int H4toH5image_attr_index(hid_t h4toh5id, int32 image_id, char * h5group_full_path, char *
h5dset_name, int attr_index)
```

**Purpose:**

Converts an HDF4 image object attribute to an HDF5 dataset attribute.

**Description:**

H4toH5image\_attr\_index converts an HDF4 image attribute, specified by its HDF4 index, to an HDF5 dataset attribute.

This function assumes the preexistence of an HDF5 dataset that the converted image attribute will belong to. If h5dset\_name is set to NULL, the HDF4 image specified in ri\_id should be converted before this function is called.

The attr\_index parameter specifies which image attribute is to be converted.

This function works for the following types of HDF4 images: GR, RI8, and RI24.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>int32</i> image_id	IN: HDF4 raster image identifier
<i>char</i> * h5group_full_path	IN: The absolute pathname of HDF5 group. Must start with "/". The converted image attribute will be associated with a member of this group as specified in h5dset_name.
<i>char</i> * h5dset_name	IN: The relative pathname of an HDF5 dataset, relative to h5group_full_path. If set to NULL, the H4toH5 library will generate a default HDF5 dataset name based on the HDF4 image name.
<i>int</i> attr_index	IN: The HDF4 index of the image attribute. Must be a value from 0 (zero), for the first image attribute, to n-1, for the last image attribute, where n is the number of HDF4 image attributes.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5pal

**Signature:**

```
int H4toH5pal(hid_t h4toh5id, int32 ri_id, char * h5group_full_path, char * h5dset_name, char
* h5pal_group_full_path, char * h5pal_name, int attr_flag, int ref_flag)
```

**Purpose:**

Converts an HDF4 palette object to an HDF5 dataset and optionally attaches it to an HDF5 image dataset.

**Description:**

H4toH5pal converts an HDF4 palette object to an HDF5 palette dataset.

The parameter `ref_flag` specifies whether to attach the palette to an HDF5 dataset. This HDF5 dataset should either be a converted HDF4 image object or conform to the HDF5 image specification (See section 3.4 and Figure 2 in *Mapping HDF4 Objects to HDF5 Objects* [1] for details.)

The `attr_flag` parameter specifies whether the palette attributes are to be converted.

**Parameters:**

<code>hid_t h4toh5id</code>	IN: h4toh5 identifier
<code>int32 ri_id</code>	IN: Raster image identifier with which the current palette is associated in the HDF4 file
<code>char * h5group_full_path</code>	IN: The absolute pathname of target HDF5 parent group of the HDF5 image dataset. Must start with "/". If <code>ref_flag</code> is set to <code>H425_NOREF</code> , this parameter will be ignored.
<code>char * h5dset_name</code>	IN: The relative pathname of an HDF5 dataset, relative to <code>h5group_full_path</code> . This HDF5 dataset should either be a converted HDF4 image object or conform to the HDF5 image specification. An object reference to the palette will be generated for this dataset. If <code>h5dset_name</code> is set to <code>NULL</code> , no object reference attribute will be generated to associate the converted palette with any HDF5 dataset.
<code>char * h5pal_group_full_path</code>	IN: The absolute pathname of the target HDF5 group to which the converted palette dataset will be assigned. Must start with "/" unless set to <code>NULL</code> . If set to <code>NULL</code> , the default palette group name <code>/HDF4_PALGROUP</code> will be used.
<code>char * h5pal_name</code>	IN: The relative pathname of the target HDF5 dataset, relative to <code>h5pal_group_full_path</code> . The palette is converted to this dataset. If <code>h5pal_name</code> is set to <code>NULL</code> , a default palette dataset name will be created as the string <code>HDF4_PALETTE_ref</code> , where <code>ref</code> is the object reference number of the palette object in the HDF4 file.

*int* attr\_flag

IN: A flag specifying whether palette attributes are to be converted:

H425\_NOATTRS (or 0)

No palette attributes are converted.

H425\_ALLATTRS (or 1)

All palette attributes are converted to attributes of the corresponding HDF5 dataset.

*int* ref\_flag

IN: A flag specifying whether the converted palette is to be associated with another HDF5 dataset:

H425\_NOREF (or 0)

The palette is not associated with another HDF5 dataset.

H425\_REF (or 1)

The palette is associated with another HDF5 dataset.

**Returns:**

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

**Non-C API(s):**

---

## Lone objects

---

**Name:** H4toH5all\_lone\_sds

**Signature:**

```
int H4toH5all_lone_sds(hid_t h4toh5id, char *h5group_full_path, char *
h5dim_group_full_path, int dim_flag, int attr_flag)
```

**Purpose:**

Converts all independent HDF4 SDS objects to HDF5 datasets.

**Description:**

H4toH5all\_lone\_sds identifies all SDS objects in the HDF4 file that are not members of a user-defined Vgroup structure and converts them to HDF5 datasets. Such SDSs are sometimes referred to as lone or independent SDSs.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>char *</i> h5group_full_path	IN: The absolute pathname of an HDF5 group. Must start with "/". The converted SDSs will become members of this group.
<i>char *</i> h5dim_group_full_path	IN: The absolute pathname of the HDF5 group to which the converted dimension scale datasets associated with the SDS will be assigned. Must start with "/". If set to NULL, the default dimension scale group name /HDF4_DIMGROUP will be used.
<i>int</i> dim_flag	IN: A flag specifying whether dimension scale datasets associated with the SDS are to be converted: 0 Dimension scale datasets are not converted. 1 Dimension scale datasets are converted.
<i>int</i> attr_flag	IN: A flag specifying whether SDS attributes are to be converted to attributes of the corresponding HDF5 datasets: H425_NOATTRS (or 0) No attributes are converted. H425_ALLATTRS (or 1) All attributes are converted. If dim_flag is set to 1, attributes of the dimension scale datasets are also converted.

**Returns:**

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

**Non-C API(s):**

**Name:** H4toH5all\_lone\_image

**Signature:**

```
int H4toH5all_lone_image(hid_t h4toh5id, char * h5group_full_path, char *
h5pal_group_full_path, int pal_flag, int attr_flag)
```

**Purpose:**

Converts all independent HDF4 image objects attribute to HDF5 datasets.

**Description:**

H4toH5all\_lone\_image identifies all image objects in the HDF4 file that are not members of a user-defined Vgroup structure and converts them to HDF5 datasets. Such images are sometimes referred to as lone or independent images.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>char</i> * h5group_full_path	IN: The absolute pathname of an HDF5 group. Must start with "/". The converted image datasets will become members of this group.
<i>char</i> * h5pal_group_full_path	IN: The absolute pathname of the HDF5 group to which the palettes associated with the converted image will be assigned. If set to NULL, the default dimension group name /HDF4_PALGROUP will be used.
<i>int</i> pal_flag	IN: A flag specifying whether palettes associated with the converted images are to be converted: H425_NOPAL Palettes are not converted. H425_PAL Palette are converted
<i>int</i> attr_flag	IN: A flag specifying whether attributes associated with the converted images are to be converted to the attributes of the corresponding HDF5 datasets: H425_NOATTRS (or 0) No attributes are converted. H425_ALLATTRS (or 1) All attributes are converted. If pal_flag is set to 1, palette attributes are also converted.

**Returns:**

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

**Non-C API(s):**



**Name:** H4toH5all\_lone\_vdata

**Signature:**

*int* H4toH5all\_lone\_vdata(*hid\_t* h4toh5id, *char* \* h5group\_full\_path, *int* attr\_flag)

**Purpose:**

Converts all lone HDF4 Vdata objects into HDF5 datasets.

**Description:**

H4toH5all\_lone\_vdata identifies all Vdatas in the HDF4 file that are not members of a user-defined Vgroup structure and converts them to HDF5 datasets.

**Parameters:**

<i>hid_t</i> h4toh5id	IN: h4toh5 identifier
<i>char</i> * h5group_full_path	IN: The absolute pathname of HDF5 group. Must start with "/". The converted Vdata datasets will become members of this group.
<i>int</i> attr_flag:	IN: A flag specifying whether Vdata attributes are to be converted: H425_NOATTRS (or 0) No attributes are converted. H425_ALLATTRS (or 1) All attributes are converted to attributes of the corresponding HDF5 datasets.

**Returns:**

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

**Non-C API(s):**

---

THG Help Desk: [help@hdfgroup.org](mailto:help@hdfgroup.org)

Describes H4H5TOOLS Conversion Library and Utilities Release 2.2.3, September 2017.