H4toH5 Conversion Library
API Reference Manual

with

H4/H5 Command-line Conversion Utilities

The HDF Group

https://www.hdfgroup.org/

Table of Contents

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.

Alphabetical Listing

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
h4toh5	Use the HDF5 dimension scale APIs, Not add the H4toH5 predefined attributes.
h4toh5 -pa	Use the HDF5 dimension scale APIs, Not add the H4toH5 predefined attributes.
h4toh5 -ospe	Follow the older version(version 3 and before) of the H4toH5 mapping document.
h4toh5 -na or h4toh5 -osna	Follow the older version of the mapping document, Not add the H4toH5 predefined attributes.
h4toh5 -sv	Map one field vdata 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:

HDF4 Object	Resulting HDF5 Object
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 file `INSTALL_Unix.txt` in the distributed "h4toh5 Conversion Software" source code package. The file is also available here in the distributed source code:

`<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 thieir 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:

char * H4file_name	IN: HDF4 filename, the source file
char * H5file_name	IN: HDF5 filename, the target file
int access	IN: The file access flag for the HDF5 file. May contain any one of the following values: H425_CREATE (or 1) Create a new HDF5 file; if the file already exists, the function will fail. H425_OPEN (or 3) Open an existing HDF5 file with read and write permission; if the file does not already exist, create a new HDF5 file. H425_CLOBBER (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 access contains any value other than the above, H425_CLOBBER 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:

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

hid_t h4toh5id	IN: h4toh5 identifier
int32 sds_id	IN: SDS identifier with which the SDS dimension scale dataset is associated in the HDF4 file
char * 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 "/".
char * 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 h5dset_name is set to NULL, no object reference attribute will be generated to associate the converted dimension scale with any HDF5 dataset.

char * `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.
char * `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.
int `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.
int `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.
int `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 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:

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

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:

hid_t `h4toh5id`	IN: h4toh5 identifier
char * `anno_label_name`	IN: The attribute name in the HDF5 file of the converted file label.
int `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:

hid_t h4toh5id	IN: h4toh5 identifier
char * anno_desc_name	IN: The attribute name in the HDF5 file of the converted file description annotation
int 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:

hid_t h4toh5id	IN: h4toh5 identifier
char * 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 h5dset_name.
char * h5dset_name	IN: The relative pathname of the target HDF5 dataset, relative to h5group_full_path. The object attribute will be associated with this dataset; if the object attribute is 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 annotation is associated.
int32 obj_tag	IN: The HDF4 tag of the HDF4 object with which the object label annotation is associated.
char * anno_label_name	IN: The name of the HDF5 attribute containing the converted object label.
int 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:

hid_t h4toh5id	IN: h4toh5 identifier
char * h5group_full_path	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 h5dset_name.
char * h5dset_name	IN: The relative pathname of the target HDF5 dataset, relative to h5group_full_path. The object attribute will be associated with this dataset; if the object attribute is 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 description annotation is associated.
int32 obj_tag	IN: The HDF4 tag of the HDF4 object with which the object description annotation is associated.
char * anno_descnname	IN: The name of the HDF5 attribute containing the converted object description.
int desc_index	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_DESCRIPITION_*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:

hid_t h4toh5id	IN: h4toh5 identifier
char * 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.
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 description annotations are associated.
int32 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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 vgroup_id	IN: Vgroup identifier in the HDF4 file
char * h5parent_group_full_path	IN: The absolute pathname of the HDF5 group of which the converted Vgroup is to be a member. Must start with "/".
char * 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 the HDF4 Vgroup name.
int vg_flag	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.
int attr_flag	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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 vgroup_id	IN: HDF4 Vgroup identifier
char * 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.
char * 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.
char * 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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 vgroup_id	IN: HDF4 Vgroup identifier
char * 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.
char * 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.
int 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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 vgroup_id	IN: Vgroup identifier in the HDF4 file
char * h5parent_group_full_path	IN: The absolute pathname of the group of which the converted Vgroup is to be a member. Must start with "/".
char * 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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 vdata_id	IN: Vdata identifier in the HDF4 file
char * h5group_full_path	IN: The absolute pathname of the HDF5 group to which the converted dataset will be assigned. Must start with "/".
char * h5dset_name	IN: The relative pathname of the target HDF5 dataset, relative to h5group_full_path. The Vdata is converted to this dataset. If h5dset_name is set to NULL, the H4toH5 library will generate a default HDF5 dataset name based on the Vdata name.
int 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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 vdata_id	IN: HDF4 Vdata identifier
char * 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.
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 Vdata name.
char * 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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 vdata_id	IN: HDF4 Vdata identifier
char * 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.
char * 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.
int 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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 vdata_id	IN: HDF4 Vdata identifier
char * 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.
char * 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.
char * field_name	IN: The HDF4 name of the Vdata field to be converted.
char * 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* 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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 vdata_id	IN: HDF4 Vdata identifier
char * 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.
char * 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.
int 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.
int 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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 sds_id	IN: SDS identifier in the HDF4 file
char * h5group_full_path	IN: The absolute pathname of the HDF5 group into which the converted SDS is to be placed. Must start with "/".
char * h5dset_name	IN: The relative pathname of the HDF5 dataset, relative to h5group_full_path. If set to NULL, the H4toH5 library generates a default HDF5 dataset name based on the SDS name.
char * 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 /HDF4_DIMGROUP is used.
int 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.
int 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 dim_flag 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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 sds_id	IN: HDF4 SDS identifier
char * 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.
char * 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.
char * 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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 sds_id	IN: HDF4 SDS identifier
char * 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 h5dset_name.
char * 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 SDS name.
int 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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 ri_id	IN: Raster image identifier in the HDF4 file
char * h5group_full_path	IN: The absolute pathname of the HDF5 group to which the converted image dataset will be assigned. Must start with "/".
char * h5dset_name	IN: The relative pathname of an HDF5 dataset, relative to h5group_full_path. If h5dset_name is set to NULL, the H4toH5 library will generate a default HDF5 dataset name based on the name of the converted image.
char * h5pal_group_full_path	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 /HDF4_PALROUP will be used.
char * h5pal_name	IN: The relative pathname of the target HDF5 dataset, relative to h5pal_group_full_path. The palette will be converted to this dataset. If h5_palname is set to NULL, a default palette dataset name is created as the string HDF4_PALETTE_*ref*, where *ref* is the reference number of the palette object in the HDF4 file.
int pal_flag	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:

hid_t `h4toh5id`	IN: h4toh5 identifier
int32 `image_id`	IN: HDF4 raster image identifier
char * `h5group_full_path`	IN: The absolute pathname of HDF5 group. Mst start with "/". The converted image attribute will be associated with a member 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.
int `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:

hid_t h4toh5id	IN: h4toh5 identifier
int32 ri_id	IN: Raster image identifier with which the current palette is associated in the HDF4 file
char * h5group_full_path	IN: The absolute pathname of target HDF5 parent group of the HDF5 image dataset. Must start with "/". If ref_flag is set to H425_NOREF, this parameter will be ignored.
char * h5dset_name	IN: The relative pathname of an HDF5 dataset, relative to h5group_full_path. 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 h5dset_name is set to NULL, no object reference attribute will be generated to associate the converted palette with any HDF5 dataset.
char * h5pal_group_full_path	IN: The absolute pathname of the target HDF5 group to which the converted palette dataset will be assigned. Must start with "/" unless set to NULL. If set to NULL, the default palette group name /HDF4_PALGROUP will be used.
char * h5pal_name	IN: The relative pathname of the target HDF5 dataset, relative to h5pal_group_full_path. The palette is converted to this dataset. If h5pal_name is set to NULL, a default palette dataset name will be created as the string HDF4_PALETTE_*ref*, where *ref* 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:

hid_t h4toh5id	IN: h4toh5 identifier
char * h5group_full_path	IN: The absolute pathname of an HDF5 group. Must start with "/". The converted SDSs will become members of this group.
char * 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.
int 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.
int 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:

hid_t h4toh5id	IN: h4toh5 identifier
char * h5group_full_path	IN: The absolute pathname of an HDF5 group. Must start with "/". The converted image datasets will become members of this group.
char * 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.
int 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
int 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:

hid_t `h4toh5id`	IN: h4toh5 identifier
char * `h5group_full_path`	IN: The absolute pathname of HDF5 group. Must start with "/". The converted Vdata datasets will become members of this group.
int `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):