vfadtr

Vaddtagref

```
int32 Vaddtagref(int32 vgroup_id, int32 tag, int32 ref)
```

vgroup_id	IN:	The Vgroup identifier returned by **Vattach**
tag	IN:	Tag of object to add
ref	IN:	Reference number of object to add

Purpose This routine inserts the tag and reference number of a data object into the specified Vgroup.

Return value Returns the number of tag and reference numbers in the Vgroup if successful and `FAIL (-1)` otherwise.

Description This routine is primarily used to add an HDF object that is neither a Vgroup nor a Vdata into a Vgroup. The HDF object is specified by the *tag* and *ref* parameters.

FORTRAN

```
integer function vfadtr(vgroup_id, tag, ref)
    integer*4 vgroup_id, tag, ref
```

Vattach

vfatch

Vattach

```
int32 Vattach(int32 file_id, int32 vgroup_ref, char *access)
```

file_id	IN:	File identifier returned by **Hopen**
vgroup_ref	IN:	Reference number for the Vgroup to open
access	IN:	Type of access; "r" for read, "w" for write.

Purpose — This routine attaches to an existing Vgroup or creates a new Vgroup.

Return value — Returns a Vgroup identifier if successful and `FAIL(-1)` otherwise.

Description — The *file_id* is the file identifier of an opened HDF file. The *vgroup_ref* parameter specifies which Vgroup in the HDF file to attach to:

if *vgroup_ref* = -1, a new Vgroup is created.
if *vgroup_ref* is positive, the Vgroup corresponding to the *vgroup_ref* is attached.

Vattach returns a Vgroup identifier for the accessed Vgroup. This identifier or vgroup_id, is used for all further operations on the Vgroup. Once operations are compelte, the vgroup_id must be disposed of via a call to **Vdetach**. Multiple attaches may be made to the same Vgroup simultaneously, giving rise to several Vgroup identifiers for the same Vgroup. Each vgroup_id must be disposed of independently.

FORTRAN

```
integer function vfatch(file_id, vgroup_ref, access)
    integer*4       file_id, vgroup_ref
    character*1     access
```

Vdetach

```
VOID Vdetach(int32 vgroup_id)
```

vgroup_id	IN:	The Vgroup identifier returned by **Vattach**

Purpose — This access routine detaches a Vgroup that is currently attached, terminating further access to that Vgroup.

Return value — None.

Description — All space associated with that Vgroup will be freed. Each attached Vgroup should be detached using this routine before the file is closed. **Vdetach** also updates the Vgroup information in the HDF file if any changes occur. The identifier *vgroup_id* should not be used after that Vgroup is detached.

FORTRAN

```
subroutine vfdtch(vgroup_id)
    integer*4 vgroup_id
```

Vend

vfend

Vend

```
VOID Vend(int32 file_id)
```

file_id	IN:	File identifier returned by **Hopen**

Purpose — Releases the internal data structures for the specified HDF file.

Return value — None.

Description — **Vend** releases all internal data structures allocated by calling **Vstart**. This routine must be called after all Vdata/Vgroup operations on an HDF file are completed. Further attempts to use Vdata or Vgroup routines after calling **Vend** will cause errors.

FORTRAN

```
subroutine vfend(file_id)
    integer*4 file_id
```

vfgcls

Vgetclass

```
VOID Vgetclass(int32 vgroup_id, char *vgroup_class)
```

vgroup_id	IN:	The Vgroup identifier returned by **Vattach**
vgroup_class	OUT:	Buffer for storing the Vgroup class name

Purpose — This routine returns the class name (if any) of a Vgroup .

Return value — None.

Description — This routine retrieves the class name of the Vgroup specified by *vgroup_id* and returns it in the buffer pointed to by *vgroup_class*. The maximum length of the name is defined by the macro VGNAMELENMAX.

FORTRAN

```
subroutine vfgcls(vgroup_id, vgroup_class)
    integer*4       vgroup_id
    character*(*)   vgroup_class
```

Vgetid

vfgid

Vgetid

```
int32 Vgetid(int32 file_id, int32 vgroup_ref)
```

file_id	IN:	File identifier returned by **Hopen**
vgroup_ref	IN:	Reference number of the previous Vgroup

Purpose
This search routine searches through the HDF file and returns the reference number of the next Vgroup following the Vgroup that has the reference number *vgroup_ref*. It is generally used to search the file sequentially for Vgroups.

Return value
Returns the reference number of the next Vdata if successful and FAIL(-1) otherwise.

Description
To initiate a search, call this routine with a vgroup_ref value of -1. This will return the reference number of the first Vgroup in the file. Searching past the last Vgroup in the file will return an error.

Example
This sample code prints out the reference numbers of all Vgroups in the HDF file.

```
int32 file_id; /* id of an opened HDF file */
int32 vgroup_ref;

file_id = Hopen("myfile.hdf",DFACC_READ,0);
vgroup_ref = -1;
while(1) {
   vgroup_ref = Vgetid(file_id, vgroup_ref);
   if (vgroup_ref == -1) break;
   printf("found Vgroup with id %d\n", vgroup_ref);
}
Hclose(file_id);
```

FORTRAN

```
integer function vfgid(file_id, vgroup_ref)
   integer*4 file_id, vgroup_ref
```

vfgnam

Vgetname

```
VOID Vgetname(int32 vgroup_id, char *vgroup_name)
```

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
vgroup_name	OUT:	Buffer to store Vgroup name

Purpose — This inquiry routine returns the name of the Vgroup.

Return value — None.

Description — This routine retrieves the name of the Vgroup specified by *vgroup_id* and returns it in the buffer pointed to by *vgroup_name*. The maximum length of the name is defined by the macro VGNAMELENMAX.

FORTRAN

```
subroutine vfgnam(vgroup_id, vgroup_name)
   integer*4      vgroup_id
   character*(*)  vgroup_name
```

Vgettagref

vfgttr

Vgettagref

```
intn Vgettagref(int32 vgroup_id, int32 index, int32 *tag, int32 *ref)
```

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
index	IN:	Index of the tag and reference number to retrieve from the Vgroup
tag	OUT:	Pointer to location to store the tag found in the indexed position
ref	OUT:	Pointer to location to store the ref number found in the indexed position

Purpose

This inquiry routine returns the tag/ref of an HDF object at a given position with a Vgroup. Do not confuse with **Vgettagrefs**.

Return value

Returns `SUCCEED(0)` if successful and `FAIL(-1)` otherwise.

Description

The input parameter *index* specifies the position within the Vgroup. This routine provides a means of accessing the tag/ref pairs in a Vgroup by specifying the position within the Vgroup.

Example

This sample code first uses **Vntagrefs** to determine the number of tag/ref pairs in a Vgroup. It then uses **Vgettagref** to extract each tag/ref one at a time.

```
int32 vgroup_id;  /* pointer to an attached Vgroup */
int32 tag, ref;
int32 status, i, npairs;

npairs = Vntagrefs(vgroup_id);
for (i=0; i<npairs; i++) {
   status = Vgettagref (vgroup_id, i, &tag, &ref);
   printf ("found tag=%d ref=%d at position %d\n", tag, ref, i);
   }
```

FORTRAN

```
integer function vfgttr(vgroup_id, tag_array, ref_array, index)
   integer*4      vgroup_id, index
   integer*4      tag, ref
```

Vgettagrefs

```
int32 Vgettagrefs(int32 vgroup_id, int32 tag_array[], int32 ref_array[],
          int32 maxsize)
```

vgroup_id	IN:	The Vgroup identifier returned by **Vattach**
tag_array	OUT:	Array to store returned tags
ref_array	OUT:	Array to store returned reference numbers
maxsize	IN:	Maximum number of tag/ref pairs to store in *tag_array* and *ref_array*

Purpose

This inquiry routine returns the tag/ref pairs of the HDF objects belonging to a given Vgroup. Do not confuse with **Vgettagref**.

Return value

Returns the number of tag/ref pairs in a specified Vgroup if successful and FAIL(-1) otherwise.

Description

The input parameter *maxsize* specifies the maximum number of tag/ref pairs to be returned. The tag/ref pairs will be returned in arrays *tag_array* and *ref_array*. Each array must be at least *maxsize* in size.

Example

This sample code uses **Vntagrefs** to determine the number of tag/ref pairs in a Vgroup. It then allocates memory , and uses **Vgettagrefs** to extract the tag/ref values.

```
int32 vgroup_id;  /* pointer to an attached Vgroup */
int32 npairs, status;
int32 *tags, * refs;

npairs = Vntagrefs(vgroup_id);
tags   = (int32*) malloc (sizeof(int32) * npairs);
refs   = (int32*) malloc (sizeof(int32) * npairs);
status = Vgettagrefs(vgroup_id, tags, refs, npairs);
```

FORTRAN

```
integer function vfgttrs(vgroup_id, tag_array, ref_array, maxsize)
    integer*4      vgroup_id, maxsize
    integer*4      tag_array(*), ref_array(*)
```

Vinqtagref

vfinqtr

Vinqtagref

```
intn Vinqtagref(int32 vgroup_id, int32 tag, int32 ref)
```

vgroup_id	IN:	The Vgroup identifier returned by **Vattach**
tag	IN:	Tag of item to check for
ref	IN:	Reference number of items to check for

Purpose — This inquiry routine tests whether a given HDF object is linked to a given Vgroup.

Return value — Returns `TRUE(1)` if the tag and reference number are linked to the specified Vgroup, and `FALSE(0)` otherwise.

Description — The HDF object is specified by the *tag* and *ref* parameter. This routine then checks if the tag/ref pair can be found in the Vgroup *vgroup_id*.

FORTRAN

```
integer function vfinqtr(vgroup_id, tag, ref)
    integer*4 vgroup_id, tag, ref
```

vfinq

Vinquire

```
intn Vinquire(int32 vgroup_id, int32 *n_entries, char *vgroup_name)
```

vgroup_id	IN:	The Vgroup identifier returned by **Vattach**
n_entries	OUT:	Pointer to variable to store the number of entries in a Vgroup
vgroup_name	OUT:	Buffer to store the name of a Vgroup.

Purpose — This is the general Vgroup inquiry routine.

Return value — Returns `SUCCEED(0)` if successful and `FAIL(-1)` otherwise.

Description — This routine retrieves the number of entries in a Vgroup and the name of the Vgroup. The maximum length of the Vgroup's name is defined by the macro `VGNAMELENMAX`.

FORTRAN

```
integer function vfinq(vgroup_id, n_entries, vgroup_name)
    integer*4       vgroup_id, n_entries
    character*(*)   vgroup_name
```

Vinsert

vfinsrt

Vinsert

```
int32 Vinsert(int32 vgroup_id, int32 v_id)
```

vgroup_id	IN:	The Vgroup identifier returned by **`Vattach`**
v_id	IN:	Identifier for the item (Vdata or Vgroup) to insert into a Vgroup

Purpose
This routine establishes a link from a Vgroup to either another Vgroup or to a Vdata identifier.

Return value
Returns the position of the inserted Vgroup or Vdata within the Vgroup if successful and `FAIL(-1)` otherwise.

Description
Essentially, **`Vinsert`** allows Vgroups or Vdatas to be grouped together. To insert other HDF objects (ie that are not Vgroups or Vdatas) into a Vgroup, use **`Vaddtagref`**. The Vdata or the Vgroup *v_id* will be added to the Vgroup *vgroup_id*.

FORTRAN

```
integer function vfinsrt(vgroup_id, v_id)
    integer*4 vgroup_id, v_id
```

Vlone

```
int32 Vlone(int32 file_id, int32 ref_array[], int32 maxsize)
```

file_id	IN:	File identifier returned by **Hopen**
ref_array	OUT:	Buffer to store the reference numbers of lone Vgroups
maxsize	IN:	Maximum number of Vgroups to store in *ref_array*.

Purpose This inquiry routine returns an array of reference numbers for all Vgroups that are not linked to any Vgroup in the HDF file.

Return value Returns the total number of lone Vgroups in a file if successful and `FAIL(-1)` otherwise.

Definition This routine is provided for applications to locate all lone Vgroups (ie those that are not grouped with other objects) in a HDF file. Another use is to locate all Vgroups at the top of any grouping hierarchy.

The parameter *maxsize* specifies the maximum number of reference numbers to be returned. The reference numbers will be returned in the array *ref_array[]*. The array must be at least *maxsize* elements in size.

The return value from this function will be the total number of Vgroups that are not linked to any Vgroup in the file. But at most *maxsize* reference numbers will be returned in *ref_array*.

An array size of 65000 integers for *ref_array* is more than adequate. The preferred method is to use dynamic memory instead; first call **Vlone** with a value of 0 for *maxsize*, and then use the returned value to allocate memory for *ref_array* before calling **Vlone**.

Example The sample code illustrates the preferred method of using **Vlone**. The second call to **Vlone** returns the required reference numbers.

```
int32 file_id;      /* id of opened HDF file */
int32 maxsize,  status;
int32 *ref_array;

maxsize = Vlone (file_id, NULL, 0);
ref_array  = (int32*) malloc(sizeof(int32)*maxsize);
status   = Vlone (file_id, ref_array, maxsize);
```

FORTRAN

```
integer function vflone(file_id, ref_array, maxsize)
   integer*4      file_id,
   integer*4      ref_array(*), maxsize
```

Vntagrefs

vfntr

Vntagrefs

`int32 Vntagrefs(int32 vgroup_id)`

vgroup_id	IN:	The Vgroup identifier returned by **`Vattach`**

Purpose — This inquiry routine returns a count of the number of HDF objects belonging to a given Vgroup.

Return value — Returns the number of HDF elements linked to the Vgroup and or `FAIL(-1)` otherwise.

Description — The routine returns a count of the number of tag and reference number pairs stored in the given Vgroup. This routine is used together with **`Vgettagrefs`**, or in a loop with **`Vgettagref`** to look at the HDF objects linked to a given Vgroup.

FORTRAN

```
integer function vfntr(vgroup_id)
   integer*4      vgroup_id
```

vfscls

Vsetclass

```
VOID Vsetclass(int32 vgroup_id, char *vgroup_class)
```

vgroup_id	IN:	The Vgroup identifier returned by **Vattach**
vgroup_class	IN:	Name of class for Vgroup

Purpose — This specify routine sets the class name for a Vgroup.

Return value — None.

Description — Vgroups initially have a class name of NULL. The class name may be set more than once. Class names, like Vgroup names, can be of any character strings. They exist solely as meaningful labels for user applications.

FORTRAN

```
subroutine vfscls(vgroup_id, vgroup_class)
   integer*4       vgroup_id
   character*(*)   vgroup_class
```

Vsetname

vfsnam

Vsetname

```
VOID Vsetname(int32 vgroup_id, char *vgroup_name)
```

vgroup_id	IN:	The Vgroup identifier returned by **Vattach**
vgroup_name	IN:	Name for the Vgroup

Purpose — This specify routine sets the name for a Vgroup.

Return value — None.

Description — Vgroups each initially have a name of NULL, and may be renamed more than once. Note that the routine does not check for uniqueness of Vgroup names.

Vgroup names are optional, but recommended. They serve as meaningful labels for user applications. If used, they should be unique.

FORTRAN

```
subroutine vfsnam(vgroup_id, vgroup_name)
   integer*4      vgroup_id
   character*(*)  vgroup_name
```

vfstart

Vstart

```
VOID Vstart(int32 file_id)
```

file_id	IN:	File identifier returned by **Hopen**

Purpose — Initialize internal data structures for HDF files containing Vdata or Vgroups.

Return value — None.

Description — **Vstart** initializes the internal Vdata and Vgroup data structures in an HDF file. This routine must be called before any Vdata or Vgroup operation is attempted on an HDF file. **Vstart** must be called once for each file involved in the operation.

FORTRAN

```
subroutine vfstart(file_id)
    integer*4       file_id
```