Vaddtagref/vfadtr

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

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

Purpose Inserts an object into a vgroup.

Return value Returns the number of objects in the vgroup if successful and `FAIL` (or `-1`) otherwise.

Description **Vaddtagref** inserts the object identified by the parameters *tag* and *ref* into the vgroup identified by the parameter *vgroup_id*.

If an object to be inserted is a data set, duplication of the tag/reference number pair will be allowed. Otherwise, the tag/reference number pair must be unique among the elements within the vgroup or the routine will return `FAIL` (or `-1`).

FORTRAN

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

integer vgroup_id, tag, ref
```

Vattach/vfatch

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
access	IN:	Type of access

Purpose Initiates access to a new or existing vgroup.

Return value Returns the vgroup identifier (*vgroup_id*) if successful and `FAIL` (or `-1`) otherwise.

Description **Vattach** opens a vgroup with access type specified by the parameter *access* in the file identified by the parameter *file_id*. The vgroup is identified by the reference number, *vgroup_ref*.

Vattach returns the vgroup identifier, *vgroup_id*, for the accessed vgroup. The *vgroup_id* is used for all subsequent operations on this vgroup. Once operations are complete, the vgroup identifier must be disposed of via a call to **Vdetach**. Multiple attaches may be made to the same vgroup simultaneously, and several vgroup identifiers can be created for the same vgroup. Each vgroup identifier must be disposed of independently.

The parameter *file_id* is the file identifier of an opened file. The parameter *vgroup_ref* specifies which vgroup in the file to attach to. If *vgroup_ref* is set to `-1`, a new vgroup will be created. If *vgroup_ref* is set to a positive number, the vgroup with that as a reference number is attached.

Possible values for the parameter *access* are “r” for read access and “w” for write access.

FORTRAN

```
integer function vfatch(file_id, vgroup_ref, access)

integer file_id, vgroup_ref

character*1 access
```

Vattrinfo/vfainfo

intn Vattrinfo(int32 *vgroup_id*, intn *attr_index,* char **attr_name,* int32 **data_type,* int32 **count,* int32 **size*)

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
attr_index	IN:	Index of the attribute
attr_name	OUT:	Name of the attribute
data_type	OUT:	Data type of the attribute
count	OUT:	Number of values in the attribute
size	OUT:	Size, in bytes, of the attribute values.

Purpose Retrieves the name, data type, number of values, and value size of an attribute for a vgroup.

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

Description **Vattrinfo** retrieves the name, datatype, number of values, and value size of an attribute identified by its index, *attr_index,* in the vgroup, *vgroup_id*. Name, data type, number of values and size are retrieved into the parameters *attr_name, data_type, count*, and *size*, respectively.

If the attribute's name, data type, number of values, or value size are not needed, the corresponding output parameters can be set to `NULL`.

The valid value *attr_index* range from 0 to the total number of attributes attached to a vgroup - 1. The number of vgroup attributes can be obtained using **Vnattrs**.

FORTRAN

```
integer function vfainfo(vgroup_id, attr_index, attr_name,
                         data_type, count, size)

integer vgroup_id, attr_index, data_type, count, size

character*(*) attr_name
```

Vdelete/vdelete

int32 Vdelete(int32 *file_id*, int32 *vgroup_id*)

file_id	IN:	File identifier returned by **Hopen**
vgroup_id	IN:	Vgroup identifier returned by **Vattach**

Purpose Remove a vgroup from a file.

Return value Returns `SUCCEED` (or `0`) if successful and `FAIL` (or `-1`) if not successful.

Description **Vdelete** removes the vgroup identified by the parameter *vgroup_id* from the file identified by the parameter *file_id*.

This routine will remove the vgroup from the internal data structures and from the file.

FORTRAN

```
integer function vdelete(file_id, vgroup_id)

integer file_id, vgroup_id
```

Vdeletetagref/vfdtr

int32 Vdeletetagref(int32 *vgroup_id*, int32 *tag*, int32 *ref*)

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

Purpose Deletes an object from a vgroup.

Return value Returns `SUCCEED` (or `0`) if successful and `FAIL` (or `-1`) if not successful or the given tag/reference number pair is not found in the vgroup.

Description **Vdeletetagref** deletes the object specified by the parameters *tag* and *ref* from the vgroup identified by the parameter *vgroup_id*. **Vinqtagref** should be used to check if the tag/reference number pair exists before calling this routine.

If duplicate tag/reference number pairs are found in the vgroup, **Vdeletetagref** deletes the first occurrence. **Vinqtagref** should be used to determine if duplicate tag/reference number pairs exist in the vgroup.

FORTRAN

```
integer function vfdtr(vgroup_id, tag, ref)

integer vgroup_id, tag, ref
```

Vdetach/vfdtch

int32 Vdetach(int32 *vgroup_id*)

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

Purpose Terminates access to a vgroup.

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

Description **Vdetach** detaches the currently-attached vgroup identified by *vgroup_id* and terminates access to that vgroup.

All space associated with the vgroup, *vgroup_id*, will be freed. Each attached vgroup must be detached by calling 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 the vgroup is detached.

FORTRAN

```
integer function vfdtch(vgroup_id)

integer vgroup_id
```

Vend/vfend

intn Vend(int32 *file_id*)

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

Purpose Terminates access to a vgroup and/or vdata interface.

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

Description **Vend** terminates access to the vgroup and/or vdata interfaces initiated by **Vstart** and all internal data structures allocated by **Vstart**.

Vend must be called after all vdata and vgroup operations on the file *file_id* are completed. Further attempts to use vdata or vgroup routines after calling **Vend** will result in a `FAIL` (or `-1`) being returned.

FORTRAN

```
integer function vfend(file_id)

integer file_id
```

Vfind/vfind

int32 Vfind(int32 *file_id*, char **vgroup_name*)

file_id	IN:	File identifier returned by **Hopen**
vgroup_name	IN:	Name of the vgroup

Purpose Returns the reference number of a vgroup given its name.

Return value Returns the reference number of the vgroup if successful and `0` otherwise.

Description **Vfind** searches the file identified by the parameter *file_id* for a vgroup with the name specified by the parameter *vgroup_name*, and returns the corresponding reference number.

If more than one vgroup has the same name, **Vfind** will return the reference number of the first one.

FORTRAN

```
integer function vfind(file_id, vgroup_name)


integer file_id

character*(*) vgroup_name
```

Vfindattr/vffdatt

intn Vfindattr(int32 *vgroup_id*, char **attr_name*)

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
attr_name	IN:	Name of the attribute

Purpose Returns the index of a vgroup attribute given its name.

Return value Returns the index of an attribute if successful and `FAIL` (or `-1`) otherwise.

Description **Vfindattr** searches the vgroup identified by the parameter *vgroup_id* for the attribute with the name specified by the parameter *attr_name*, and returns the index of that attribute.

If more than one attribute has the same name, **Vfindattr** will return the index of the first one.

FORTRAN

```
integer function vffdatt(vgroup_id, attr_name)

integer vgroup_id

character*(*) attr_name
```

Vfindclass/vfndcls

int32 Vfindclass(int32 *file_id*, char **vgroup_class*)

file_id	IN:	File identifier returned by **Hopen**
vgroup_class	IN:	Class name of the vgroup

Purpose Returns the reference number of a vgroup specified by its class name.

Return value Returns the reference number of the vgroup if successful and `0` otherwise.

Description **Vfindclass** searches the file identified by the parameter *file_id* for the vgroup with the class name specified by the parameter *vgroup_class*, and returns the reference number of that vgroup.

If more than one vgroup has the same class name, **Vfindclass** will return the reference number of the first one.

FORTRAN

```
integer function vfndcls(file_id, vgroup_class)

integer file_id

character*(*) vgroup_class
```

Vflocate/vffloc

int32 Vflocate(int32 *vgroup_id*, char **field_name*)

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
field_name_list	IN:	List of field names

Purpose Locates a vdata in a vgroup given a list of field names.

Return value Returns the reference number of the vdata if successful and `FAIL` (or `-1`) otherwise.

Description **Vflocate** searches the vgroup identified by the parameter *vgroup_id* for a vdata that contains all of the fields listed in the parameter *field_name_list*. If that vdata is found, **Vflocate** will return its reference number.

FORTRAN

```
integer function vffloc(vgroup_id, field_name)

integer vgroup_id

character*(*) field_name
```

Vgetattr/vfgnatt/vfgcatt

intn Vgetattr(int32 *vgroup_id*, intn *attr_index,* VOIDP *attr_values*)

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
attr_index	IN:	Index of the attribute
attr_values	OUT:	Buffer for the attribute values

Purpose Retrieves the values of a vgroup attribute.

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

Description **Vgetattr** retrieves the values of the attribute identified by its index, *attr_index*, into the buffer *attr_values* for the vgroup identified by the parameter *vgroup_id*.

The valid values of the parameter *attr_index* range from 0 to the total number of vgroup attributes - 1. The total number of attributes can be obtained using **Vnattrs**. To determine the amount of memory sufficient to hold the attribute values, the user can obtain the number of attribute values and the attribute value size using **Vattrinfo**.

FORTRAN

```
integer function vfgnatt(vgroup_id, attr_index, attr_values)

integer vgroup_id, attr_index

<valid numeric data type> attr_values

integer function vfgcatt(vgroup_id, attr_index, attr_values)

integer vgroup_id, attr_index

character*(*) attr_values
```

Vgetclass/vfgcls

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

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
vgroup_class	OUT:	Class name of the vgroup

Purpose Retrieves the class name of a vgroup.

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

Description **Vgetclass** retrieves the class name of the vgroup identified by the parameter *vgroup_id* in the buffer *vgroup_class*.

The maximum length of the name is defined by `VGNAMELENMAX (or 64)`.

FORTRAN

```
integer function vfgcls(vgroup_id, vgroup_class)

integer vgroup_id

character*(*) vgroup_class
```

Vgetid/vfgid

int32 Vgetid(int32 *file_id*, int32 *vgroup_ref*)

file_id	IN:	File identifier returned by **Hopen**
vgroup_ref	IN:	Reference number of the current vgroup

Purpose Returns the reference number of the next vgroup.

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

Description **Vgetid** sequentially searches the file identified by the parameter *file_id* and returns the reference number of the vgroup following the vgroup that has the reference number specified by the parameter *vgroup_ref*.

The search is initiated by calling 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 cause **Vgetid** to return `FAIL` (or `-1`).

FORTRAN

```
integer function vfgid(file_id, vgroup_ref)

integer file_id, vgroup_ref
```

Vgetname/vfgnam

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

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
vgroup_name	OUT:	Name of the vgroup

Purpose Retrieves the name of a vgroup.

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

Description **Vgetname** retrieves the name of the vgroup identified by the parameter *vgroup_id* into the buffer *vgroup_name*. The maximum length of the name is defined by `VGNAMELENMAX` (or `64`).

FORTRAN

```
integer function vfgnam(vgroup_id, vgroup_name)

integer vgroup_id

character*(*) vgroup_name
```

Vgetnext/vfgnxt

int32 Vgetnext(int32 *vgroup_id*, int32 *v_ref*)

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
v_ref	IN:	Reference number of the vgroup or vdata

Purpose Gets the reference number of the next member (vgroup or vdata only) of a vgroup.

Return value Returns the reference number of the vgroup or vdata if successful and `FAIL` (or `-1`) otherwise.

Description **Vgetnext** searches in the vgroup identified by the parameter *vgroup_id* for the object following the object specified by its reference number *v_ref*. Either of the two objects can be a vgroup or a vdata. If *v_ref* is set to -1, the routine will return the reference number of the first vgroup or vdata in the vgroup.

Note that this routine only gets a vgroup or a vdata in a vgroup. **Vgettagrefs** gets any object in a vgroup.

FORTRAN

```
integer function vfgnxt(vgroup_id, v_ref)


integer vgroup_id, v_ref
```

Vgettagref/vfgttr

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

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
index	IN:	Index of the object in the vgroup
tag	OUT:	Tag of the object
ref	OUT:	Reference number of the object

Purpose Retrieves the tag/reference number pair of an object given its index within a vgroup.

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

Description **Vgettagref** retrieves the tag/reference number pair of the object specified by its index, *index,* within the vgroup identified by the parameter *vgroup_id*. Note that this routine is different from **Vgettagrefs**, which retrieves the tag/reference number pairs of a number of objects.

The valid values of *index* range from 0 to the total number of objects in the vgroup - 1. The total number of objects in the vgroup can be obtained using **Vinquire**.

The tag is stored in the buffer *tag* and the reference number is stored in the buffer *ref*.

FORTRAN

```
integer function vfgttr(vgroup_id, index, tag, ref)

integer vgroup_id, index

integer tag, ref
```

Vgettagrefs/vfgttrs

int32 Vgettagrefs(int32 *vgroup_id*, int32 *tag_array*[], int32 *ref_array*[], int32 *num_of_pairs*)

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
tag_array	OUT:	Array of tags
ref_array	OUT:	Array of reference numbers
num_of_pairs	IN:	Number of tag/reference number pairs

Purpose Retrieves the tag/reference number pairs of the HDF objects belonging to a vgroup.

Return value Returns the number of tag/reference number pairs obtained from a vgroup if successful and `FAIL` (or `-1`) otherwise.

Description **Vgettagrefs** retrieves at most *num_of_pairs* number of tag/reference number pairs belonging to the vgroup, *vgroup_id*, and stores them in the buffers *tag_array* and *ref_array*.

The input parameter *num_of_pairs* specifies the maximum number of tag/reference number pairs to be returned. The size of the arrays, *tag_array* and *ref_array*, must be at least *num_of_pairs*.

FORTRAN

```
integer function vfgttrs(vgroup_id, tag_array, ref_array,
                                    num_of_pairs)

integer vgroup_id, num_of_pairs

integer tag_array(*), ref_array(*)
```

Vgetversion/vfgver

int32 Vgetversion(int32 *vgroup_id*)

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

Purpose Gets the version of a vgroup.

Return value Returns the vgroup version number if successful, and `FAIL` (or `-1`) otherwise.

Description **Vgetversion** returns the version number of the vgroup identified by the parameter *vgroup_id*. There are three valid version numbers: `VSET_OLD_VERSION` (or `2`), `VSET_VERSION` (or `3`), and `VSET_NEW_VERSION` (or `4`).

`VSET_OLD_VERSION` is returned when the vgroup is of a version that corresponds to an HDF library version before version 3.2.

`VSET_VERSION` is returned when the vgroup is of a version that corresponds to an HDF library version between versions 3.2 and 4.0 release 2.

`VSET_NEW_VERSION` is returned when the vgroup is of the version that corresponds to an HDF library version of version 4.1 release 1 or higher.

FORTRAN

```
integer function vfgver(vgroup_id)

integer vgroup_id
```

Vinqtagref/vfinqtr

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

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

Purpose Checks whether an object belongs to a vgroup.

Return value Returns `TRUE` (or `1`) if the object belongs to the vgroup, and `FALSE` (or `0`) otherwise.

Description **Vinqtagref** checks if the object identified by its tag, *tag,* and its reference number, *ref,* belongs to the vgroup identified by the parameter *vgroup_id*.

FORTRAN

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


integer vgroup_id, tag, ref
```

Vinquire/vfinq

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

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
n_entries	OUT:	Number of entries in a vgroup
vgroup_name	OUT:	Name of a vgroup

Purpose Retrieves the number of entries in a vgroup and its name.

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

Description **Vinquire** retrieves the name of and the number of entries in the vgroup identified by the parameter *vgroup_id* into the buffer *vgroup_name* and the parameter *n_entries*, respectively.

The maximum length of the vgroup name is defined by `VGNAMELENMAX` (or `64`).

FORTRAN

```
integer function vfinq(vgroup_id, n_entries, vgroup_name)

integer vgroup_id, n_entries

character*(*) vgroup_name
```

Vinsert/vfinsrt

int32 Vinsert(int32 *vgroup_id*, int32 *v_id*)

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
v_id	IN:	Identifier of the vdata or vgroup

Purpose Inserts a vdata or vgroup into a vgroup.

Return value Returns the position (*index*) of the inserted element within the vgroup if successful and `FAIL` (or `-1`) otherwise.

Description **Vinsert** inserts the vdata or vgroup identified by the parameter *v_id* into the vgroup identified by the parameter *vgroup_id*.

Essentially, **Vinsert** only inserts a vgroup or vdata. To insert any objects into a vgroup, use **Vaddtagref**.

The returned value, *index*, is either 0 or a positive value, which indicates the position of the inserted element in the vgroup.

FORTRAN

```
integer function vfinsrt(vgroup_id, v_id)


integer vgroup_id, v_id
```

Visvg/vfisvg

intn Visvg(int32 *vgroup_id*, int32 *obj_ref*)

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
obj_ref	IN:	Reference number of the object

Purpose Determines whether an element of a vgroup is a vgroup and a member of another vgroup.

Return value Returns `TRUE` (or `1`) if the object is a vgroup and `FALSE` (or `0`) otherwise.

Description **Visvg** determines if the object specified by the reference number, *obj_ref,* is a vgroup within the vgroup identified by the parameter *vgroup_id*.

FORTRAN

```
integer function vfisvg(vgroup_id, obj_ref)

integer vgroup_id, obj_ref
```

Visvs/vfisvs

intn Visvs(int32 *vgroup_id*, int32 *obj_ref*)

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
obj_ref	IN:	Reference number of the object

Purpose Determines whether a data object is a vdata within a vgroup.

Return value Returns `TRUE` (or `1`) if the object is a vdata and `FALSE` (or `0`) otherwise.

Description **Visvs** determines if the object specified by the reference number, *obj_ref,* is a vdata within the vgroup identified by the parameter *vgroup_id*.

FORTRAN

```
integer function vfisvs(vgroup_id, obj_ref)

integer vgroup_id, obj_ref
```

Vlone/vflone

int32 Vlone(int32 *file_id*, int32 *ref_array*[], int32 *max_refs*)

file_id	IN:	File identifier returned by **Hopen**
ref_array	OUT:	Array of reference numbers
max_refs	IN:	Maximum number of lone vgroups to be retrieved

Purpose Retrieves the reference numbers of lone vgroups, i.e., vgroups that are at the top of the grouping hierarchy, in a file.

Return value Returns the total number of lone vgroups if successful and `FAIL` (or `-1`) otherwise.

Description **Vlone** retrieves the reference numbers of lone vgroups in the file identified by the parameter *file_id*. Although **Vlone** returns the total number of lone vgroups in the file, only at most *max_refs* reference numbers are retrieved and stored in the buffer *ref_array*. The array must have at least *max_refs* elements.

An array size of 65,000 integers for *ref_array* is more than adequate if the user chooses to declare the array statically. However, the preferred method is to dynamically allocate memory instead; first call **Vlone** with a value of `0` for *max_refs*, and then use the returned value to allocate memory for *ref_array* before calling **Vlone** again.

FORTRAN

```
integer function vflone(file_id, ref_array, max_refs)


integer file_id, ref_array(*), max_refs
```

Vnattrs/vfnatts

intn Vnattrs(int32 *vgroup_id*)

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

Purpose Returns the number of attributes assigned to a vgroup.

Return value Returns the total number of attributes assigned to the specified vgroups if successful and `FAIL` (or `-1`) otherwise.

Description **Vnattrs** gets the number of attributes assigned to the vgroup identified by the parameter *vgroup_id*.

FORTRAN

```
integer function vfnatts(vgroup_id)

integer vgroup_id
```

Vnrefs/vnrefs

int32 Vnrefs(int32 *vgroup_id*, int32 *tag_type*)

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
tag_type	IN:	Type of the tag

Purpose Returns the number of tags of a given tag type in a vgroup.

Return value Returns 0 or the total number of tags if successful and `FAIL` (or `-1`) otherwise.

Description **Vnrefs** returns 0 or the number of tags having the type specified by the parameter *tag_type* in the vgroup identified by the parameter *vgroup_id*.

See Appendix A, *NCSA HDF Tags*, in the *HDF User's Guide*, for a discussion of tag types.

FORTRAN

```
integer function vnrefs(vgroup_id, tag_type)


integer vgroup_id, tag_type
```

Vntagrefs/vfntr

int32 Vntagrefs(int32 *vgroup_id*)

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

Purpose Returns the number of objects in a vgroup.

Return value Returns `0` or a positive number representing the number of HDF objects linked to the vgroup if successful or `FAIL` (or `-1`) otherwise.

Description **Vntagrefs** returns the number of objects in a vgroup identified by the parameter *vgroup_id*.

Vntagrefs is used together with **Vgettagrefs**, or with **Vgettagref** to look at the data objects linked to a given vgroup.

FORTRAN

```
integer function vfntr(vgroup_id)

integer vgroup_id
```

Vsetattr/vfsnatt/vfscatt

intn Vsetattr(int32 *vgroup_id*, char **attr_name,* int32 *data_type,* int32 *count,* VOIDP *values*)

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
attr_name	IN:	Name of the attribute
data_type	IN:	Data type of the attribute
count	IN:	Number of values the attribute contains
values	IN:	Buffer containing the attribute values

Purpose Attaches an attribute to a vgroup.

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

Description **Vsetattr** attaches an attribute to the vgroup identified by the parameter *vgroup_id*. The attribute name is specified by the parameter *attr_name* and the attribute data type is specified by the parameter *data_type*. The values of the attribute are specified by the parameter *values*, and the number of values in the attribute is specified by the parameter *count*. Refer to Table 1A in Section I of this manual for a listing of all valid data types.

If the attribute already exists, the new values will replace the current ones, provided the data type and the number of attribute values have not been changed. If either the data type or the order have been changed, **Vsetattr** will return `FAIL` (or `-1`).

FORTRAN

```
integer vfsnatt(vgroup_id, attr_name, data_type, count, values)

integer vgroup_id, data_type, count
<valid numeric data type> values(*)
character*(*) attr_name
```

```
integer vfscatt(vgroup_id, attr_name, data_type, count, values)

integer vgroup_id, data_type, count
character*(*) attr_name, values(*)
```

Vsetclass/vfscls

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

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
vgroup_class	IN:	Class name of a vgroup

Purpose Sets the class name of a vgroup.

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

Description **Vsetclass** sets the class name specified by the parameter *vgroup_class* to the vgroup identified by the parameter *vgroup_id*.

A vgroup initially has 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.

The class name is limited to `VSNAMELENMAX` (or `64`) characters.

FORTRAN

```
integer function vfscls(vgroup_id, vgroup_class)

integer vgroup_id

character*(*) vgroup_class
```

Vsetname/vfsnam

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

vgroup_id	IN:	Vgroup identifier returned by **Vattach**
vgroup_name	IN:	Name of a vgroup

Purpose Sets the name of a vgroup.

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

Description **Vsetname** sets the name specified by the parameter *vgroup_name* for the vgroup identified by the parameter *vgroup_id*.

A vgroup initially has a name of `NULL`, and may be renamed more than once during the scope of the vgroup identifier (*vgroup_id*). 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. The name length is limited to `VSNAMELENMAX` (or `64`) characters.

FORTRAN

```
integer function vfsnam(vgroup_id, vgroup_name)

integer vgroup_id

character*(*) vgroup_name
```

Vstart/vfstart

intn Vstart(int32 *file_id*)

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

Purpose Initializes the vdata and/or vgroup interface.

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

Description **Vstart** initializes the vdata and/or vgroup interfaces for the file identified by the parameter *file_id*.

Vstart 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

```
integer function vfstart(file_id)

integer file_id
```

VHmakegroup/vhfmkgp

int32 VHmakegroup(int32 *file_id*, int32 *tag_array*[], int32 *ref_array*[], int32 *n_objects*, char **vgroup_name*, char **vgroup_class*)

file_id	IN:	File identifier returned by **Hopen**
tag_array	IN:	Array of tags
ref_array	IN:	Array of reference numbers
n_objects	IN:	Number of data objects to be stored
vgroup_name	IN:	Name of the vgroup
vgroup_class	IN:	Class of the vgroup

Purpose Creates a vgroup.

Return value Returns the reference number of the newly-created vgroup if successful, `FAIL` (or `-1`) otherwise.

Description **VHmakegroup** creates a vgroup with the name specified by the parameter *vgroup_name* and the class name specified by the parameter *vgroup_class* in the file identified by the parameter *file_id*. The routine inserts *n_objects* objects into the vgroup. The tag and reference numbers of the objects to be inserted are specified in the arrays *tag_array* and *ref_array*.

Creating empty vgroups with **VHmakegroup** is allowed. **VHmakegroup** does not check if the tag/reference number pair is valid, or if the corresponding data object exists. However, all of the tag/reference number pairs must be unique.

Vstart must precede any calls to **VHmakegroup**. It is not necessary, however, to call **Vattach** or **Vdetach** in conjunction with **VHmakegroup**.

The elements in the arrays *tag_array* and *ref_array* are the matching tag/reference number pairs of the objects to be inserted, that means *tag_array*[`0`] and *ref_array*[`0`] refer to one data object, and *tag_array*[`1`] and *ref_array*[`1`] to another, etc.

FORTRAN

```
integer function vhfmkgp(file_id, tag_array, ref_array, n_objects,
                         vgroup_name, vgroup_class)

integer file_id, n_objects

character*(*) vgroup_name, vgroup_class

integer tag_array(*), ref_array(*)
```

VQueryref/vqref

int32 VQueryref(int32 *vgroup_id*)

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

Purpose Returns the reference number of a vgroup.

Return value Returns the reference number if successful, and `FAIL` (or `-1`) otherwise.

Description **VQueryref** returns the reference number of the vgroup identified by the parameter *vgroup_id*.

FORTRAN

```
integer function vqref(vgroup_id)

integer vgroup_id
```

VQuerytag/vqtag

int32 VQuerytag(int32 *vgroup_id*)

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

Purpose Returns the tag of a vgroup.

Return value Returns the tag if successful, and `FAIL` (or -1) otherwise.

Description **VQuerytag** returns the tag of the vgroup identified by the parameter *vgroup_id*.

FORTRAN

```
integer function vqtag(vgroup_id)

integer vgroup_id
```

VFfieldesize/vffesiz

int32 VFfieldesize(int32 *vdata_id,* int32 *field_index*)

vdata_id	IN:	Vdata identifier returned by **VSattach**
field_index	IN:	Vdata field index

Purpose Returns the size, as stored on disk, of a vdata field.

Return value Returns the vdata field size if successful and `FAIL` (or `-1`) otherwise.

Description **VFfieldesize** returns the size, as stored on disk, of a vdata field identified by the parameter *field_index* in the vdata identified by the parameter *vdata_id*.

The value of the parameter *field_index* ranges from 0 to the total number of fields in the vdata - 1. The number of vdata fields is returned by **VFnfields** function.

FORTRAN

```
integer function vffesiz(vdata_id, field_index)


integer vdata_id, field_index
```

VFfieldisize/vffisiz

int32 VFfieldisize(int32 *vdata_id*, int32 *field_index*)

vdata_id	IN:	Vdata identifier returned by **VSattach**
field_index	IN:	Vdata field index

Purpose Returns the size, as stored in memory, of a vdata field.

Return value Returns the vdata field size if successful and `FAIL` (or `-1`) otherwise.

Description **VFfieldisize** returns the size, as stored in memory, of a vdata field identified by the parameter *field_index* in the vdata identified by the parameter *vdata_id*.

The value of the parameter *field_index* ranges from 0 to the total number of fields in the vdata - 1. The number of vdata fields is returned by **VFnfields** function.

FORTRAN

```
integer function vffisiz(vdata_id, field_index)

integer vdata_id, field_index
```

VFfieldname/vffname

char *VFfieldname(int32 *vdata_id*, int32 *field_index*)

vdata_id	IN:	Vdata identifier returned by **VSattach**
field_index	IN:	Vdata field index

Purpose Returns the name of a vdata field.

Return value Returns a pointer to the vdata field name if successful and `NULL` otherwise. The FORTRAN-77 version of this routine, **vffname**, returns `SUCCEED` (or `0`) or `FAIL` (or `-1`).

Description **VFfieldname** returns the name of the vdata field identified by the parameter *field_index* in the vdata identified by the parameter *vdata_id*.

The value of the parameter *field_index* ranges from 0 to the total number of fields in the vdata - 1. The number of vdata fields is returned by **VFnfields** function.

The FORTRAN-77 version of this routine, **vffname**, returns the field name in the parameter *fname*.

FORTRAN

```
integer function vffname(vdata_id, field_index, fname)

integer vdata_id, field_index

character*(*) fname
```

VFfieldorder/vffordr

int32 VFfieldorder(int32 *vdata_id*, int32 *field_index*)

vdata_id	IN:	Vdata identifier returned by **VSattach**
field_index	IN:	Vdata field index

Purpose Returns the order of a vdata field.

Return value Returns the order of the field if successful and `FAIL` (or `-1`) otherwise.

Description **VFfieldorder** returns the order of the vdata field identified by its index, *field_index,* in the vdata identified by the parameter *vdata_id*.

The value of the parameter *field_index* ranges from 0 to the total number of fields in the vdata - 1. The number of vdata fields is returned by **VFnfields** function.

FORTRAN

```
integer function vffordr(vdata_id, field_index)

integer vdata_id, field_index
```

VFfieldtype/vfftype

int32 VFfieldtype(int32 *vdata_id*, int32 *field_index*)

vdata_id	IN:	Vdata identifier returned by **VSattach**
field_index	IN:	Vdata field index

Purpose Returns the data type of a vdata field.

Return value Returns the data type if successful and `FAIL` (or `-1`) otherwise.

Description **VFfieldtype** returns the data type of the vdata field identified by its index, *field_index,* in the vdata identified by the parameter *vdata_id*.

The value of the parameter *field_index* ranges from 0 to the total number of fields in the vdata - 1. The number of vdata fields is returned by **VFnfields** function.

FORTRAN

```
integer function vfftype(vdata_id, field_index)

integer vdata_id, field_index
```

VFnfields/vfnflds

int32 VFnfields(int32 *vdata_id*)

vdata_id IN: Vdata identifier returned by **VSattach**

Purpose Returns the total number of fields in a vdata.

Return value Returns the total number of fields if successful and `FAIL` (or `-1`) otherwise.

Description **VFnfields** returns the total number of fields in the vdata identified by the parameter *vdata_id*.

FORTRAN

```
integer function vfnflds(vdata_id)

integer vdata_id
```