

Vaddtagref/vfadtr

Vaddtagref/vfadtr

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

<i>vgroup_id</i>	IN: Vgroup identifier returned by Vattach
<i>tag</i>	IN: Tag of object to add
<i>ref</i>	IN: Reference number of object to add

Purpose This routine inserts the tag/reference number pairs 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` (or `-1`) otherwise.

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

FORTTRAN

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

integer vgroup_id, tag, ref
```

Vattach/vfatch

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

<i>file_id</i>	IN: File identifier returned by Hopen
<i>vgroup_ref</i>	IN: Reference number for the vgroup to open
<i>access</i>	IN: Type of access - "r" for read, "w" for write.

Purpose	Attaches to an existing vgroup or creates a new vgroup and returns an identifier for it.
----------------	--

Return value	Returns a vgroup identifier if successful and <code>FAIL</code> (or <code>-1</code>) otherwise.
---------------------	--

Description	The <i>file_id</i> is the file identifier of an opened HDF file. The <i>vgroup_ref</i> parameter specifies which vgroup in the HDF file to attach to: if <i>vgroup_ref</i> = -1, a new vgroup is created. if <i>vgroup_ref</i> is positive, the vgroup corresponding to the <i>vgroup_ref</i> is attached. Vattach returns a vgroup identifier for the accessed vgroup. This identifier or <i>vgroup_id</i> , is used for all further operations on the vgroup. Once operations are complete, the <i>vgroup_id</i> 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 <i>vgroup_id</i> must be disposed of independently.
--------------------	---

FORTTRAN	<pre>integer function vfatch(file_id, vgroup_ref, access) integer file_id, vgroup_ref character access</pre>
-----------------	---

Vattrinfo/vfainfo

Vattrinfo/vfainfo

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

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

Purpose Returns the name, data type, number of values, and the size of the values of the specified attribute of the specified vgroup identified by *vgroup_id* and *attr_index*.

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

Description The values of the *attr_name*, *data_type*, *count* and *size* parameters can be set to NULL, if the information returned by these parameters are not needed.

The value of *attr_index* parameter is used as the index of the target vgroup attribute, and is zero-based. For example, a *attr_index* value of 4 would refer to the fifth attribute of the vgroup.

FORTTRAN

```
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
```

Vdetach/vfdtch

int32 Vdetach(int32 *vgroup_id*)

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

Purpose Detaches a currently-attached vgroup, terminating further access to that vgroup.

Return value None.

Description All space associated with the specified vgroup will be freed. Each attached vgroup should 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 that vgroup is detached.

FORTRAN integer function vfdtch(*vgroup_id*)
 integer *vgroup_id*

Vend/vfend

Vend/vfend

intn 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 and vgroup operations on an HDF file are completed. Further attempts to use vdata or vgroup routines after calling **Vend** will result in an error condition.

FORTRAN integer function vfend(*file_id*)

 integer *file_id*

Vfind/vfind

int32 Vfind(int32 *file_id*, const char **vgroupname*)

<i>file_id</i>	IN: File identifier returned by Hopen
<i>vgroupname</i>	IN: Name of the vgroup

Purpose	Returns the reference number of the vgroup with the specified name.
Return value	Returns the reference number of the vgroup if successful and 0 otherwise.

FORTTRAN	integer function vfind(<i>file_id</i> , <i>vgroupname</i>) integer <i>file_id</i> character* (*) <i>vgroupname</i>
----------	--

Vfindattr/vffdatt

Vfindattr/vffdatt

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

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

attr_name OUT: Name of the target attribute

Purpose Returns the index of an attribute with the given name, specified by the value of the *attr_name* parameter, of a vgroup.

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

FORTRAN integer function vffdatt(*vgroup_id*, *attr_name*)

 integer *group_id*
 character* (*) *attr_name*

Vfindclass/vfndcls

int32 Vfindclass(int32 *file_id*, const char **vgclass*)

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

vgclass IN: Class of the vgroup

Purpose Returns the reference number of the vgroup with the specified class.

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

FORTRAN integer function vfndcls(*file_id*, *vgclass*)

integer *file_id*
character* (*) *vgclass*

Vflocate/vffloc

Vflocate/vffloc

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

vgroup_id IN: Vgroup identifier

field IN: Target field name

Purpose Determines if the specified field name exists in the vgroup referred to by *vgroup_id*.

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

FORTTRAN integer function vffloc(*vgroup_id*, *field*)

integer *vgroup_id*
character* (*) *field*

Vgetattr/vfgnatt/vfgcatt

intn Vgetattr(int32 *vgroup_id*, intn *attr_index*, char **values*)

<i>vgroup_id</i>	IN: Vgroup identifier returned by Vattach
<i>attr_index</i>	IN: Index of the target attribute
<i>values</i>	OUT: Name of the target attribute

Purpose Returns all of the values of the specified attribute of the specified vgroup.

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

Description The *attr_index* parameter is the ordinal, zero-based index of the target attribute.

FORTRAN

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

integer function vfgcatt(vgroup_id, attr_index, values)

integer vgroup_id, attr_index
character* (*) values
```

Vgetclass/vfgcls

Vgetclass/vfgcls

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

<i>vgroup_id</i>	IN: Vgroup identifier returned by Vattach
<i>vgroup_class</i>	OUT: Buffer for storing the vgroup class name

Purpose	Retrieves the class name of the vgroup specified by <i>vgroup_id</i> and returns it in the buffer pointed to by <i>vgroup_class</i> .
----------------	---

Return value	None.
---------------------	-------

Description	The maximum length of the name is defined by the macro <code>VGNAMELENMAX</code> .
--------------------	--

FORTTRAN	<pre>integer function vfgcls(vgroup_id, vgroup_class) integer vgroup_id character* (*) vgroup_class</pre>
-----------------	--

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 previous vgroup

Purpose 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 used to sequentially search the file for vgroups.

Return value Returns the reference number of the next vdata if successful and `FAIL` (or `-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);
Vstart(file_id);
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);
}

Vend(file_id);
Hclose(file_id);
```

FORTRAN integer function vfgid(file_id, vgroup_ref)

 integer file_id, vgroup_ref

Vgetname/vfgnam

Vgetname/vfgnam

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

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

vgroup_name OUT: Buffer for the vgroup name

Purpose 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 VGNAMLENMAX.

FORTTRAN integer function vfgnam(*vgroup_id*, *vgroup_name*)

integer *vgroup_id*
character* (*) *vgroup_name*

Vgetnext/vfgnxt

int32 Vgetnext(int32 *vgroup_id*, int32 *elem_ref*)

<i>vgroup_id</i>	IN: Vgroup identifier
<i>elem_ref</i>	IN: Reference number of the vgroup or vdata

Purpose	Searches in the vgroup specified by <i>vgroup_id</i> for the vgroup or vdata following the vgroup or vdata referred to by <i>elem_ref</i> .
Return value	Returns the reference number of the target vgroup or vdata if successful and <code>FAIL</code> (or -1) otherwise.
Description	<p>If <i>elem_ref</i> is set to -1 the reference number of the first element in the vgroup is returned.</p> <p>Vgetnext will only examine vset elements. To examine all element links in a vgroup, use Vgettagrefs.</p>

FORTTRAN	<pre>integer function vfgnxt(vgroup_id, elem_ref) integer vgroup_id, elem_ref</pre>
----------	---

Vgettagref/vfgttr

Vgettagref/vfgttr

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

<i>vgroup_id</i>	IN: Vgroup identifier returned by Vattach
<i>index</i>	IN: Index of the tag/reference number pair to be retrieved from the vgroup
<i>tag</i>	OUT: Buffer for the returned tag
<i>ref</i>	OUT: Buffer for the returned reference number

Purpose Returns the tag/reference number pair of an HDF object at a given position within a vgroup. Do not confuse this routine with **Vgettagrefs**.

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

Description The input parameter *index* specifies the position within the vgroup. This routine provides a means of accessing the tag/reference number pairs in a vgroup by specifying the position within the vgroup.

Example This sample code first uses **Vntagrefs** to determine the number of tag/reference number pairs in a vgroup. It then uses **Vgettagref** to extract each tag/reference number pair 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);
}
```

FORTTRAN

```
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 maxsize)
```

<i>vgroup_id</i>	IN: Vgroup identifier returned by Vattach
<i>tag_array</i>	OUT: Buffer for the returned tags
<i>ref_array</i>	OUT: Buffer for the returned reference numbers
<i>maxsize</i>	IN: Maximum number of tag/reference number pairs to store in <i>tag_array</i> and <i>ref_array</i>

Purpose	Returns the tag/reference number pairs of the HDF objects belonging to a given vgroup. Do not confuse this routine with Vgettagref .
Return value	Returns the number of tag/reference number pairs in a specified vgroup if successful and <code>FAIL</code> (or <code>-1</code>) otherwise.
Description	The input parameter <i>maxsize</i> specifies the maximum number of tag/reference number pairs to be returned. The tag/reference number pairs will be returned in arrays <i>tag_array</i> and <i>ref_array</i> . Each array must be at least <i>maxsize</i> in size.
Example	This sample code uses Vntagrefs to determine the number of tag/reference number pairs in a vgroup. It then allocates memory, and uses Vgettagrefs to extract the tag/reference number pair 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);
```

FORTTRAN	<pre>integer function vfgttrs(vgroup_id, tag_array, ref_array, maxsize) integer vgroup_id, maxsize integer tag_array(*), ref_array(*)</pre>
-----------------	--

Vgetversion/vfgver

Vgetversion/vfgver

int32 Vgetversion(int32 *vgroup_id*)

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

Purpose **Vgetversion** returns a value of `VSET_NEW_VERSION` when encountering a vgroup of this vgroup version.

Return value Returns the vset version number if successful, and `FAIL` otherwise.

Description Version 3 is the vset version corresponding to all versions of the HDF library between 3.2 and 4.0 release 2. **Vgetversion** returns a value of `VSET_VERSION` when encountering a vgroup of this vset version. The third version is version 2. This vset version corresponds to all HDF library versions before version 3.2, and **Vgetversion** returns a value of `VSET_OLD_VERSION` when encountering a vgroup of this vset version.

FORTRAN integer function vfgver(*vgroup_id*)

 integer *vgroup_id*

Vinqtagref/vfinqtr

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

<i>vgroup_id</i>	IN: Vgroup identifier returned by Vattach
<i>tag</i>	IN: Tag of object to be checked for
<i>ref</i>	IN: Reference number of object to be checked for

Purpose Tests whether a given HDF object is linked to a given vgroup.

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

Description The HDF object is specified by the *tag* and *ref* parameters. This routine then checks if the tag/reference number pair can be found in the vgroup specified by *vgroup_id*.

FORTTRAN

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

Vinquire/vfinq

Vinquire/vfinq

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

<i>vgroup_id</i>	IN: Vgroup identifier returned by Vattach
<i>n_entries</i>	OUT: Pointer to variable to store the number of entries in a vgroup
<i>vgroup_name</i>	OUT: Buffer to store the name of a vgroup.

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

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

Description The maximum length of the vgroup's name is defined by `VGNAMELENMAX`.

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*)

<i>vgroup_id</i>	IN: Vgroup identifier returned by Vattach
<i>v_id</i>	IN: Identifier for the item, vdata or vgroup, to insert into a vgroup

Purpose	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 <code>FAIL</code> (or <code>-1</code>) otherwise.
---------------------	---

Description	Essentially, Vinsert allows vgroups or vdatas to be grouped together. To insert other HDF objects (i.e., that are not vgroups or vdatas) into a vgroup, use Vaddtagref . The vdata or the vgroup <i>v_id</i> will be added to the vgroup <i>vgroup_id</i> .
--------------------	---

FORTTRAN	<code>integer function vfinsrt(vgroup_id, v_id)</code> <code>integer vgroup_id, v_id</code>
----------	--

Visvg/vfisvg

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

<i>vgroup_id</i>	IN: Vgroup identifier returned by Vattach
<i>v_ref</i>	IN: Vgroup or vdata reference number to be queried

Purpose Determines whether the vgroup or vdata specified by *v_ref* is contained within the vgroup identified by *vgroup_id*.

Return value Returns `TRUE` (or 1) if the reference number specified by the *v_ref* parameter refers to an object stored in the specified vgroup and `FAIL` (or -1) otherwise.

FORTTRAN

```
integer function vfisvg(vgroup_id, v_ref)

integer vgroup_id, v_ref
```

Visvs/vfisvs

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

<i>vgroup_id</i>	IN:	Vgroup identifier returned by Vattach
<i>vdata_ref</i>	IN:	Vdata reference number to be queried

Purpose	Determines whether the vdata specified by <i>vdata_ref</i> is contained within another the vgroup identified by <i>vgroup_id</i> .
----------------	--

Return value	Returns TRUE (or 1) if the vdata reference number specified by the <i>vdata_ref</i> parameter is that of an object stored in the specified vgroup and FAIL (or -1) otherwise.
---------------------	---

Description	Generally called after Vgettagrefs to determine which entries are vdatas.
--------------------	--

FORTRAN	<pre>integer function vfisvs(vgroup_id, vdata_ref) integer vgroup_id, vdata_ref</pre>
----------------	---

Vlone/vflone

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

<i>file_id</i>	IN: File identifier returned by Hopen
<i>ref_array</i>	OUT: Buffer for the reference numbers of the lone vgroups
<i>maxsize</i>	IN: Maximum number of vgroups to store in <i>ref_array</i> .

Purpose 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` (or `-1`) otherwise.

Description This routine is provided for applications to locate all lone vgroups (i.e., those that are not grouped with other objects) in a HDF file. Another use is to locate all vgroups at the top of the 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 65,000 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 target 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);
```

FORTTRAN

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

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

Vnattrsvfnatts

intn Vnattrsvfnatts(int32 vgroup_id)

vgroup_id IN: Vgroup identifier returned by Vattach

Purpose Returns the number of attributes assigned to the vgroup identified by vgroup_id.

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

FORTRAN integer function vfnatts(vgroup_id)
integer group_id

Vnrefs/vnrefs

Vnrefs/vnrefs

int32 Vnrefs(int32 *vgroup_id*, int32 *tag*)

<i>vgroup_id</i>	IN: Vgroup identifier
<i>tag</i>	IN: Target tag definition

Purpose Returns the number of tags of the type specified by *tag* in the vgroup referred to by *vgroup_id*.

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

FORTRAN `integer function vnrefs(vgroup_id, tag)`

 `integer vgroup_id, tag`

Vntagrefs/vntrc

int32 Vntagrefs(int32 *vgroup_id*)

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

Purpose Returns a count of the number of tag and reference number pairs stored in the given vgroup.

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

Description This routine is used together with **Vgettagrefs**, or in a loop with **Vgettagref** to look at the HDF objects linked to a given vgroup.

FORTTRAN integer function vntrc(vgroup_id)

 integer vgroup_id

Vsetattr/vfsnatt/vfscatt

Vsetattr/vfsnatt/vfscatt

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

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

Purpose	Attaches an attribute to a vgroup.
Return value	Returns <code>SUCCESS</code> (or 0) if successful and <code>FAIL</code> (or -1) otherwise.
Description	If the attribute already exists, the new values will replace the current ones, provided the data type and order have not been changed. If either the data type or the order have been changed, Vsetattr will exit on an error condition.

FORTRAN

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

integer vgroup_id, data_type, count
<valid numerical 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, const char *vgroup_class)
```

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

vgroup_class IN: Name of class for vgroup

Purpose Sets the class name for a vgroup.

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

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.

FORTTRAN `integer function vfscls(vgroup_id, vgroup_class)`

```
integer vgroup_id  
character* (*) vgroup_class
```

Vsetname/vfsnam

Vsetname/vfsnam

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

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

vgroup_name IN: Name to be assigned to the vgroup

Purpose Sets the name for a vgroup.

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

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.

The maximum length of *vgroup_name* is `VGNAMELENMAX`. If a longer name is specified it will be truncated to that length.

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 internal vdata and vgroup data structures in an HDF file.

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

Description 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 integer function vfstart(file_id)

 integer file_id