ANannlen/afannlen

int32 ANannlen(int32 *ann_id*)

ann_id	IN:	Annotation identifier returned by **ANcreate**, **ANcreatef**, or **ANselect**

Purpose Returns the length of an annotation.

Return value Returns the length of the annotation or `FAIL` (or `-1`) otherwise.

Description **ANannlen** returns the number of characters contained in the annotation specified by the parameter *ann_id*. This function is commonly used to determine the size of a buffer to store the annotation upon reading.

FORTRAN

```
integer function afannlen(ann_id)

integer ann_id
```

ANannlist/afannlist

intn ANannlist(int32 *an_id*, ann_type *annot_type,* uint16 *obj_tag,* uint16 *obj_ref,* int32 **ann_list*)

an_id	IN:	AN interface identifier returned by **ANstart**
annot_type	IN:	Type of the annotation
obj_tag	IN:	Tag of the object
obj_ref	IN:	Reference number of the object
ann_list	OUT:	Buffer for the annotation identifiers

Purpose Retrieves the annotation identifiers of an object.

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

Description **ANannlist** obtains a list of identifiers of the annotations that are of the type specified by the parameter *annot_type* and are attached to the object identified by its tag, *obj_tag,* and its reference number, *obj_ref*.

Since this routine is implemented only to obtain the identifiers of data annotations and not file annotations, the valid values of *annot_type* are `AN_DATA_LABEL` (or `0`) and `AN_DATA_DESC` (or `1`). To obtain file annotation identifiers, use **ANfileinfo** to determine the number of file labels and descriptions, and then use **ANselect** to obtain each file annotation identifier.

Sufficient space must be allocated for *ann_list* to hold the list of annotation identifiers. This can be done by using **ANnumann** to obtain the number of annotation identifiers to be retrieved, and then allocating memory for *ann_list* using this number.

FORTRAN

```
integer function afannlist(an_id, annot_type, obj_tag, obj_ref,
                                  ann_list)

integer ann_list(*)
integer an_id, obj_tag, obj_ref, annot_type
```

ANatype2tag/afatypetag

uint16 ANatype2tag(ann_type **annot_type*)

annot_type	IN:	Type of the annotation

Purpose Returns the annotation tag corresponding to an annotation type.

Return value Returns the annotation tag (*ann_tag*) if successful, and `DFTAG_NULL` (or `0`) otherwise.

Description **ANatype2tag** returns the tag that corresponds to the annotation type specified by the parameter *annot_type*.

The following table lists the valid values of *annot_type* in the left column and the corresponding values for the returned annotation tag on the right.

Annotation Type	Annotation Tag
AN_DATA_LABEL (or 0)	DFTAG_DIL (or 104)
AN_DATA_DESC (or 1)	DFTAG_DIA (or 105)
AN_FILE_LABEL (or 2)	DFTAG_FID (or 100)
AN_FILE_DESC (or 3)	DFTAG_FD (or 101)

FORTRAN

```
integer function afatypetag(annot_type)

integer annot_type
```

ANcreate/afcreate

int32 ANcreate(int32 *an_id,* uint16 *obj_tag,* uint16 *obj_ref,* ann_type *annot_type*)

an_id	IN:	AN interface identifier returned by **ANstart**
obj_tag	IN:	Tag of the object to be annotated
obj_ref	IN:	Reference number of the object to be annotated
annot_type	IN:	Type of the data annotation

Purpose Creates a data annotation for an object.

Return value Returns the data annotation identifier (*ann_id*) if successful and `FAIL` (or `-1`) otherwise.

Description **ANcreate** creates a data annotation of type *annot_type* for the object specified by its tag, *obj_tag,* and its reference number, *obj_ref*. The returned data annotation identifier can represent either a data label or a data description.

Valid values for *annot_type* are `AN_DATA_LABEL` (or `0`) or `AN_DATA_DESC` (or `1`).

Use **ANcreatef** to create a file annotation.

Currently, the user must write to a newly-created annotation before creating another annotation of the same type. Creating two consecutive annotations of the same type causes the second call to **ANcreate** to return `FAIL` (or `-1`).

FORTRAN

```
integer function afcreate(an_id, obj_tag, obj_ref, annot_type)

integer an_id, obj_tag, obj_ref, annot_type
```

ANcreatef/affcreate

int32 ANcreatef(int32 *an_id,* ann_type *annot_type*)

an_id	IN:	AN interface identifier returned by **ANstart**
annot_type	IN:	Type of the file annotation

Purpose Creates a file annotation.

Return value Returns the file annotation identifier (*ann_id*) if successful and `FAIL` (or `-1`) otherwise.

Description **ANcreatef** creates a file annotation of the type specified by the parameter *annot_type*. The file annotation identifier returned can either represent a file label or a file description.

Valid values for *annot_type* are `AN_FILE_LABEL` (or `2`) and `AN_FILE_DESC` (or `3`).

Use **ANcreate** to create a data annotation.

Currently, the user must write to a newly-created annotation before creating another annotation of the same type. Creating two consecutive annotations of the same type causes the second call to **ANcreate** to return `FAIL` (or `-1`).

FORTRAN

```
integer function affcreate(an_id, annot_type)

integer an_id, annot_type
```

ANend/afend

int32 ANend(int32 *an_id*)

an_id	IN:	AN interface identifier returned by **ANstart**

Purpose Terminates access to an AN interface.

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

Description **ANend** terminates access to the AN interface identified by *an_id*, which is previously initialized by a call to **ANstart**. Note that there must be one call to **ANend** for each call to **ANstart**.

FORTRAN

```
integer function afend(an_id)

integer an_id
```

ANendaccess/afendaccess

intn ANendaccess(int32 *ann_id*)

ann_id	IN:	Annotation identifier returned by **ANcreate**, **ANcreatef** or **ANselect**

Purpose Terminates access to an annotation.

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

Description **ANendaccess** terminates access to the annotation identified by the parameter *ann_id*. Note that there must be one call to **ANendaccess** for every call to **ANselect**, **ANcreate** or **ANcreatef**.

FORTRAN

```
integer function afendaccess(ann_id)

integer ann_id
```

ANfileinfo/affileinfo

intn ANfileinfo(int32 *an_id*, int32 *n_*file_labels,* int32 **n_file_descs,* int32 **n_data_labels,* int32 **n_data_descs*)

an_id	IN:	AN interface identifier returned by **ANstart**
n_file_labels	OUT:	Number of file labels
n_file_descs	OUT:	Number of file descriptions
n_data_labels	OUT:	Number of data labels
n_data_descs	OUT:	Number of data descriptions

Purpose: Retrieves the number of annotations of each type in a file.

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

Description: **ANfileinfo** retrieves the total number of the four kinds of annotations and stores them in the appropriate parameters. The total number of data labels of all data objects in the file is stored in *n_data_labels*. The total number of data descriptions of all data objects in the file is stored in *n_data_descs*. The total number of file labels is stored in *n_file_labels* and the total number of file descriptions in *n_file_descs*.

Note that the numbers of data labels and descriptions refer to the total number of data labels and data descriptions in the file, not for a specific object. Use **ANnumann** to determine these numbers for a specific object.

This routine is generally used to find the range of acceptable indices for **ANselect** calls.

FORTRAN

```
integer function affileinfo(an_id, n_file_labels, n_file_descs,
                            n_data_labels, n_data_descs)

integer an_id, n_file_labels, n_file_descs

integer n_data_labels, n_data_descs
```

ANget_tagref/afgettagref

int32 ANget_tagref(int32 *an_id*, int32 *index,* ann_type *annot_type*, uint16 **ann_tag*, uint16 **ann_ref*)

an_id	IN:	AN interface identifier returned by **ANstart**
index	IN:	Index of the annotation
annot_type	IN:	Type of the annotation
ann_tag	OUT:	Tag of the annotation
ann_ref	OUT:	Reference number of the annotation

Purpose Retrieves the tag/reference number pair of an annotation given its index and type.

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

Description **ANget_tagref** retrieves the tag and reference number of the annotation identified by its index, the parameter *index*, and by its annotation type, the parameter *annot_type*. The tag is stored in the parameter *ann_tag* and the reference number is stored in the parameter *ann_ref*.

The parameter *index* is a nonnegative integer and is less than the total number of annotations of type *annot_type* in the file. Use **ANfileinfo** to obtain the total number of annotations of each type in the file.

The following table lists the valid values of the parameter *annot_type* in the left column, and the corresponding values of the parameter *ann_tag* in the right column.

Annotation Type	Annotation Tag
AN_DATA_LABEL (or 0)	**DFTAG_DIL (or 104)**
AN_DATA_DESC (or 1)	**DFTAG_DIA (or 105)**
AN_FILE_LABEL (or 2)	**DFTAG_FID (or 100)**
AN_FILE_DESC (or 3)	**DFTAG_FD (or 101)**

FORTRAN

```
integer function afgettagref(an_id, index, annot_type, ann_tag,
                                      ann_ref)

integer an_id, index, annot_type

integer ann_tag, ann_ref
```

ANid2tagref/afidtagref

int32 ANid2tagref(int32 *ann_id*, uint16 **ann_tag,* uint16 **ann_ref*)

ann_id	IN:	Annotation identifier returned by **ANselect**, **ANcreate** or **ANcreatef**
ann_tag	OUT:	Tag of the annotation
ann_ref	OUT:	Reference number of the annotation

Purpose Retrieves the tag/reference number pair of an annotation given its identifier.

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

Description **ANid2tagref** retrieves the tag/reference number pair of the annotation identified by the parameter *ann_id*. The tag is stored in the parameter *ann_tag* and the reference number is stored in the parameter *ann_ref*.

Possible values returned in *ann_tag* are `DFTAG_DIL` (or `104`) for a data label, `DFTAG_DIA` (or `105`) for a data description, `DFTAG_FID` (or `100`) for a file label and `DFTAG_FD` (or `101`) for a file description.

FORTRAN

```
integer function afidtagref(ann_id, ann_tag, ann_ref)

integer ann_id, ann_tag, ann_ref
```

ANnumann/afnumann

intn ANnumann(int32 *an_id*, ann_type *annot_type,* uint16 *obj_tag,* uint16 *obj_ref*)

an_id	IN:	AN interface identifier returned by **ANstart**
annot_type	IN:	Type of the annotation
obj_tag	IN:	Tag of the object
obj_ref	IN:	Reference number of the object

Purpose Returns the number of annotations of a given type attached to an object.

Return value Returns the number of annotations or `FAIL` (or `-1`) otherwise.

Description **ANnumann** returns the total number of annotations that are of type *annot_type* and that are attached to the object identified by its tag, *obj_tag*, and its reference number, *obj_ref*.

Since this routine is implemented only to obtain the total number of data annotations and not file annotations, the valid values of *annot_type* are `AN_DATA_LABEL` (or `0`) and `AN_DATA_DESC` (or `1`). To obtain the total number of file annotations or all data annotations, use **ANfileinfo**.

FORTRAN

```
integer function afnumann(an_id, annot_type, obj_tag, obj_ref)

integer an_id, obj_tag, obj_ref, annot_type
```

ANreadann/afreadann

int32 ANreadann(int32 *ann_id*, char* *ann_buf*, int32 *ann_length*)

ann_id	IN:	Annotation identifier returned by **ANcreate**, **ANcreatef** or **ANselect**
ann_buf	OUT:	Buffer for the annotation
ann_length	IN:	Length of the buffer *ann_buf*

Purpose Reads an annotation.

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

Description **ANreadann** reads the annotation identified by the parameter *ann_id* and stores the annotation in the parameter *ann_buf*.

The parameter *ann_length* specifies the size of the buffer *ann_buf*. If the length of the file or data label to be read is greater than or equal to *ann_length*, the label will be truncated to *ann_length* - 1 characters. If the length of the file or data description is greater than *ann_length*, the description will be truncated to *ann_length* characters. The HDF library adds a `NULL` character to the retrieved label but not to the retrieved description. The user must add a `NULL` character to the retrieved description if the C library string functions are to operate on this description.

FORTRAN

```
integer function afreadann(ann_id, ann_buf, ann_length)

integer ann_id, ann_length

character*(*) ann_buf
```

ANselect/afselect

int32 ANselect(int32 *an_id*, int32 *index*, ann_type *annot_type*)

an_id	IN:	AN interface identifier returned by **ANstart**
index	IN:	Location of the annotation in the file
annot_type	IN:	Type of the annotation

Purpose Obtains an existing annotation.

Return value Returns the annotation identifier (*ann_id*) if successful or `FAIL` (or `-1`) otherwise.

Description **ANselect** obtains the identifier of the annotation specified by its index, *index,* and by its annotation type, *annot_type*.

The parameter *index* is a nonnegative integer and is less than the total number of annotations of type *annot_type* in the file. Use **ANfileinfo** to obtain the total number of annotations of each type in the file.

Valid values of *annot_type* are `AN_DATA_LABEL` (or 0), `AN_DATA_DESC` (or 1), `AN_FILE_LABEL` (or 2), and `AN_FILE_DESC` (or 3).

FORTRAN

```
integer function afselect(an_id, index, annot_type)

integer an_id, index

integer annot_type
```

ANstart/afstart

int32 ANstart(int32 *file_id*)

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

Purpose Initializes the AN interface.

Return value Returns the AN interface identifier (*an_id*) if successful and `FAIL` (or `-1`) otherwise.

Description **ANstart** initializes the AN interface for the file identified by the parameter *file_id*. A call to **ANstart** is required before any AN functions can be invoked. **ANstart** is used with the **ANend** function to define the extent of AN interface session. A call to **ANend** is required for each call to **ANstart**.

FORTRAN

```
integer function afstart(file_id)

integer file_id
```

ANtag2atype/aftagatype

ann_type ANtag2atype(uint16 *ann_tag*)

ann_tag	IN:	Tag of the annotation

Purpose Returns the annotation type corresponding to an annotation tag.

Return value Returns the annotation type if successful or `AN_UNDEF` (or `-1`) otherwise.

Description **ANtag2atype** returns the annotation type that corresponds to the annotation tag specified by the parameter *ann_tag*.

The following table lists the valid values of *ann_tag* in the left column and the corresponding values of the returned annotation type in the right column.

Annotation Tag	Annotation Type
DFTAG_DIL (or 104)	AN_DATA_LABEL (or 0)
DFTAG_DIA (or 105)	AN_DATA_DESC (or 1)
DFTAG_FID (or 100)	AN_FILE_LABEL (or 2)
DFTAG_FD (or 101)	AN_FILE_DESC (or 3)

FORTRAN

```
integer function aftagatype(ann_tag)

integer ann_tag
```

ANtagref2id/aftagrefid

int32 ANtagref2id(int32 *an_id*, uint16 *ann_tag*, uint16 *ann_ref*)

an_id	IN:	AN interface identifier returned by **ANstart**
ann_tag	IN:	Tag of the annotation
ann_ref	IN:	Reference number of the annotation

Purpose Returns the identifier of an annotation given its tag/reference number pair.

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

Description **ANtagref2id** returns the identifier of the annotation specified by its tag, *ann_tag*, and its reference number, *ann_ref*.

Valid values of *ann_tag* are `DFTAG_DIL` (or `104`) for a data label, `DFTAG_DIA` (or `105`) for a data description, `DFTAG_FID` (or `100`) for a file label, and `DFTAG_FD` (or `101`) for a file description.

FORTRAN

```
integer function aftagrefid(an_id, ann_tag, ann_ref)

integer an_id, ann_tag, ann_ref
```

ANwriteann/afwriteann

int32 ANwriteann(int32 *ann_id*, char* *ann*, int32 *ann_length*)

ann_id	IN:	Annotation identifier returned by **ANcreate**, **ANcreatef**, or **ANselect**
ann	IN:	Text to be written to the annotation
ann_length	IN:	Length of the annotation text

Purpose Writes an annotation.

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

Description **ANwriteann** writes the annotation text provided in the parameter *ann* to the annotation specified by the parameter *ann_id*. The parameter *ann_length* specifies the number of characters in the annotation text.

If the annotation has already been written with text, **ANwriteann** will overwrite the current text.

FORTRAN

```
integer function afwriteann(ann_id, ann, ann_length)

integer ann_id, ann_length

character*(*) ann
```