

DFANaddfds/daafds

intn DFANaddfds(int32 *file_id*, char **description*, int32 *desc_len*)

<i>file_id</i>	IN: File identifier returned by Hopen
<i>description</i>	IN: Sequence of ASCII characters (may include <code>NULL</code> or <code>'\0'</code>)
<i>desc_len</i>	IN: Length of the description

Purpose Adds a file description to a file.

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

Description These annotations are associated with the file, not with any particular object within the file. The parameter *description* can contain any sequence of ASCII characters. It does not have to be a string. Use the general purpose routines **Hopen** and **Hclose** to manage file access as the file annotation routines will not open and close HDF files.

Example This example illustrates the use of **DFANaddfid** and **DFANaddfds** to write a file label and description to an HDF file.

```
#include "hdf.h"
#define MAXLABELN 80
#define MAXDESCLEN 1000
...
int32 file_id;
char label[MAXLABELN+1], description[MAXDESCLEN+1];

file_id = Hopen("myfile", DFACC_WRITE, 0);

/* store a file label in the file */
strcpy (label, "File #1");
DFANaddfid(file_id, label);

/* store description in file */
DFANaddfds(file_id, description, desc_len(description));

Hclose(file_id);
...
```

FORTTRAN	integer function daafds(file_id, description, desc_len)
	integer file_id, desc_len
	character* (*) description

DFANaddfid/daafid

```
intn DFANaddfid(int32 file_id, char *label)
```

file_id IN: The file identifier returned by **Hopen**.

label IN: A null-terminated string.

Purpose Writes a file label to a file.

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

Description These annotations are associated with the file, not with any particular object within the file. The label must be a single string. Use the general purpose routines **Hopen** and **Hclose** to manage file access because the file annotation routines will not open and close HDF files for you.

In the Fortran-77 version, the string length for the label should be close to the actual expected string length, because in Fortran-77 string lengths generally are assumed to be the declared length of the array that holds the string.

FORTTRAN `integer function daafid(file_id, label)`

 `integer file_id`
 `character* (*) label`

DFANclear/daclear

DFANclear/daclear

intn DFANclear(void)

Purpose	Resets all internal library structures and parameters of the DFAN annotation interface.
Return value	Returns <code>SUCCESS</code> (or 0) if successful and <code>FAIL</code> (or -1) otherwise.
Description	When a file is regenerated in a single run by a library routine of another interface (such as DFSDputdata), DFANclear should be called to reset the interface.
FORTRAN	<code>integer function daclear()</code>

DFANgetdesc/dagdesc

```
intn DFANgetdesc(char *filename, uint16 tag, uint16 ref, char *desc_buf, int32 buf_len)
```

<i>filename</i>	IN: Name of the file
<i>tag</i>	IN: Tag of the data object assigned the description
<i>ref</i>	IN: Reference number of the data object assigned the description
<i>desc_buf</i>	OUT: Buffer allocated to hold the description
<i>buf_len</i>	IN: Size of the buffer allocated to hold the description

Purpose	Reads the description assigned to the data object with the given tag and reference number.
Return value	Returns <code>SUCCESS</code> (or 0) if successful and <code>FAIL</code> (or -1) otherwise.
Description	The parameter <i>buf_len</i> specifies the storage space available for the description. The length of <i>buf_len</i> must account for the null termination character appended to the description.

FORTTRAN	<pre>integer function dagdesc(filename, tag, ref, desc_buf, buf_len) character* (*) filename, desc_buf integer tag, ref integer buf_len</pre>
-----------------	--

DFANgetdesclen/dagdlen

DFANgetdesclen/dagdlen

int32 DFANgetdesclen(char **filename*, uint16 tag, uint16 *ref*)

<i>filename</i>	IN: Name of the file
<i>tag</i>	IN: Tag of the data object assigned the description
<i>ref</i>	IN: Reference number of the data object assigned the description

Purpose Retrieves the length of a description of the data object with the given tag and reference number.

Return value Returns the length of a description if successful and `FAIL` (or `-1`) otherwise.

Description This routine should be used to insure that there is enough space allocated for a description before actually reading it.

FORTRAN

```
integer function dagdlen(filename, tag, ref)

character* (*) filename
integer tag, ref
```

DFANgetfds/dagfds

int32 DFANgetfds(int32 *file_id*, char **desc_buf*, int32 *buf_len*, intn *isfirst*)

<i>file_id</i>	IN: File identifier returned by Hopen
<i>desc_buf</i>	OUT: The buffer allocated to hold the description
<i>buf_len</i>	IN: Size of the buffer allocated to hold the description
<i>isfirst</i>	IN: Determines the description to be retrieved

Purpose	Reads the next file description.
Return value	Returns the length of the file description if successful and <code>FAIL</code> (or <code>-1</code>) otherwise.
Description	<p>If <i>isfirst</i> is 0, DFANgetfds gets the next file description from an HDF file. For example, if there are three file descriptions in a file, three successive calls to DFANgetfds will get all three descriptions. If <i>isfirst</i> is 1, DFANgetfds gets the first file description.</p> <p>Valid values for <i>isfirst</i> are: 1 to read the first description and 0 to read the next description.</p>

FORTTRAN

```
integer function dagfds(file_id, desc_buf, buf_len, isfirst)

integer file_id, buf_len, isfirst
character* (*) desc_buf
```

DFANgetfdslen/dagfdsl

DFANgetfdslen/dagfdsl

int32 DFANgetfdslen(int32 *file_id*, intn *isfirst*)

<i>file_id</i>	IN: File identifier returned by Hopen
<i>isfirst</i>	IN: Determines the description the retrieved length information applies to

Purpose Returns the length of a file description.

Return value Returns the length of the file description if successful and `FAIL` (or `-1`) otherwise.

Description When **DFANgetfdslen** is first called for a given file, it returns the length of the first file description. In order to get the lengths of successive file descriptions, you must call **DFANgetfds** between calls to **DFANgetfdslen**. Successive calls to **DFANgetfdslen** without calling **DFANgetfds** between them will return the length of the same file description.

Valid values for *isfirst* are: 1 to read the length of the first description and 0 to read the length of the next description.

FORTRAN

```
integer function dagfdsl(file_id, isfirst)

integer file_id, isfirst
```

DFANgetfid/dagfid

```
int32 DFANgetfid(int32 file_id, char *desc_buf, int32 buf_len, intn isfirst)
```

<i>file_id</i>	IN: File identifier returned by Hopen
<i>label_buf</i>	OUT: The buffer allocated to hold the label
<i>buf_len</i>	IN: Size of the buffer allocated to hold the label
<i>isfirst</i>	IN: Determines the file label to be retrieved

Purpose	Reads a file label from a file.
Return value	Returns the length of the file description if successful and <code>FAIL</code> (or <code>-1</code>) otherwise.
Description	<p>If <i>isfirst</i> is 0, DFANgetfid gets the next file label from the file. If <i>isfirst</i> is 1, DFANgetfid gets the first file label in the file. If <i>buf_len</i> is not large enough, the label is truncated to <i>buf_len</i>-1 characters in the buffer <i>label_buf</i>.</p> <p>Valid values of <i>isfirst</i> are: 1 to read the first label, 0 to read the next label</p>

FORTRAN	<pre>integer function dagfid(file_id, label_buf, buf_len, isfirst) integer file_id, buf_len, isfirst character* (*) label_buf</pre>
---------	--

DFANgetfidlen/dagfidl

DFANgetfidlen/dagfidl

int32 DFANgetfidlen(int32 *file_id*, intn *isfirst*)

<i>file_id</i>	IN: File identifier returned by Hopen
<i>isfirst</i>	IN: Determines the file label the retrieved length information applies to

Purpose Returns the length of a file label.

Return value Returns the length of the file label if successful and `FAIL` (or `-1`) otherwise.

Description When **DFANgetfidlen** is first called for a given file, it returns the length of the first file label. In order to retrieve the lengths of successive file labels, **DFANgetfid** must be called between calls to **DFANgetfidlen**. Otherwise, successive calls to **DFANgetfidlen** will return the length of the same file label.

Valid values of *isfirst* are: 1 to read the first label, and 0 to read the next label.

FORTTRAN	<code>integer function dagfidl(file_id, isfirst)</code> <code>integer file_id, isfirst</code>
----------	--

DFANgetlabel/daglab

```
intn DFANgetlabel(char *filename, uint16 tag, uint16 ref, char *label_buf, int32 buf_len)
```

<i>filename</i>	IN: Name of the HDF file
<i>tag</i>	IN: Tag of the data object assigned the label
<i>ref</i>	IN: Reference number of the data object assigned the label
<i>label_buf</i>	OUT: Buffer for the label
<i>buf_len</i>	IN: Size of the buffer allocated for the label

Purpose	Reads the label assigned to the data object identified by the given tag and reference number.
Return value	Returns <code>SUCCESS</code> (or 0) if successful and <code>FAIL</code> (or -1) otherwise.
Description	The parameter <i>buf_len</i> specifies the storage space available for the label. The length of <i>buf_len</i> must account for the null termination character appended to the annotation.

FORTTRAN	<pre>integer function daglab(filename, tag, ref, label_buf, buf_len) character* (*) filename, label_buf integer tag, ref, buf_len</pre>
----------	--

DFANgetlablen/dagllen

DFANgetlablen/dagllen

int32 DFANgetlablen(char **filename*, uint16 *tag*, uint16 *ref*)

<i>filename</i>	IN: Name of the file
<i>tag</i>	IN: Tag of the data object assigned the label
<i>ref</i>	IN: Reference number the data object assigned the label

Purpose Returns the length of a label assigned to the object with a given tag and reference number.

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

Description This routine should be used to insure that there is enough space allocated for a label before actually reading it.

FORTRAN

```
integer function dagllen(filename, tag, ref)

character* (*) filename
integer tag, ref
```

DFANlablist/dallist

```
int DFANlablist(char *filename, uint16 tag, unit16 ref_list[], char *label_list, int list_len, intn
               label_len, intn start_pos)
```

<i>filename</i>	IN: Name of the file
<i>tag</i>	IN: Tag to be queried
<i>ref_list</i>	OUT: Buffer for the returned reference numbers
<i>label_list</i>	OUT: Buffer for the returned labels
<i>list_len</i>	IN: Size of the reference number list and the label list
<i>label_len</i>	IN: Maximum length allowed for a label
<i>start_pos</i>	IN: Starting position of the search

Purpose Returns a list of all reference numbers and labels (if labels exist) for a given tag.

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

Description Entries are returned from the *start_pos* entry up to the *list_len* entry.

The *list_len* determines the number of available entries in the reference number and label lists, *label_len* is the maximum length allowed for a label, and *start_pos* tells which label to start reading for the given tag. (If *start_pos* is 1, for instance, all labels will be read; if *start_pos* is 4, all but the first 3 labels will be read.) The *ref_list* contains a list of reference numbers for all objects with a given tag. The *label_list* contains a corresponding list of labels, if any. If there is no label stored for a given object, the corresponding entry in *label_list* is an empty string.

Taken together, the *ref_list* and *label_list* constitute a directory of all objects and their labels (where they exist) for a given tag. The *label_list* parameter can display all of the labels for a given tag. Or it can be searched to find the reference number of a data object with a certain label. Once the reference number for a given label is found, the corresponding data object can be accessed by invoking other HDF routines. Therefore, this routine provides a mechanism for the direct access to data objects in HDF files.

```
FORTTRAN
integer function dallist(filename, tag, ref_list, label_list,
                        list_len, label_len, start_pos)

character* (*) filename, label_list
integer ref_list(*)
integer list_len, label_len, start_pos
```

DFANlastref/dalref

DFANlastref/dalref

uint16 DFANlastref()

Purpose	Returns the reference number of the annotation last written or read.
Return value	Returns the reference number if successful and <code>FAIL</code> (or <code>-1</code>) otherwise.

FORTTRAN	<code>integer function dalref()</code>
-----------------	---

DFANputdesc/dapdesc

```
int DFANputdesc(char *filename, uint16 tag, uint16 ref, char *description, int32 desc_len)
```

<i>filename</i>	IN:	Name of the file
<i>tag</i>	IN:	Tag of the data object to be assigned the description
<i>ref</i>	IN:	Reference number the data object to be assigned the description
<i>description</i>	IN:	Sequence of ASCII characters (may include <code>NULL</code> or <code>'\0'</code>)
<i>desc_len</i>	IN:	Length of the description

Purpose	Writes a description for the data object with the given tag and reference number.
Return value	Returns <code>SUCCEED</code> (or 0) if successful and <code>FAIL</code> (or -1) otherwise.
Description	The parameter description can contain any sequence of ASCII characters; it does not have to be a string. If DFANputdesc is called more than once for the same tag/reference number pair, only the last description is stored in the file.

FORTTRAN

```
integer function dapdesc(filename, tag, ref, description,  
                        desc_len)  
  
character* (*) filename, description  
integer tag, ref, desc_len
```

DFANputlabel/daplab

DFANputlabel/daplab

intn DFANputlabel(char **filename*, uint16 *tag*, uint16 *ref*, char **label*)

<i>filename</i>	IN: Name of the file
<i>tag</i>	IN: Tag of the data object to be assigned the label
<i>ref</i>	IN: Reference number the data object to be assigned the label
<i>label</i>	IN: Null-terminated label string

Purpose Assigns a label to the data object with the given tag/reference number pair.

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

FORTRAN

```
integer function daplab(filename, tag, ref, label)

character* (*) filename, label
integer tag, ref
```