DFR8addimage/d8aimg

intn DFR8addimage(char **filename*, VOIDP *image*, int32 *width*, int32 *height*, uint16 *compress*)

filename	IN:	Name of the file
image	IN:	Array containing the image data
width	IN:	Number of columns in the image
height	IN:	Number of rows in the image
compress	IN:	Type of compression to use, if any

Purpose **DFR8addimage** appends the RIS8 for the image to the file.

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

Description **DFR8addimage** is functionally equivalent to **DFR8putimage**, except that **DFR8putimage** cannot append image data - it only overwrites.

FORTRAN

```
integer function d8aimg(filename, image, width, height,
    compress)

character* (*) filename, image
integer width, height
integer compress
```

DFR8getdims/d8gdims

intn DFR8getdims(char **filename*, int32 **width*, int32 **height*, intn **ispalette*)

filename	IN:	Name of the HDF file
width	OUT:	Number of columns in the next image in the file
height	OUT:	Number of rows in the next image in the file
ispalette	OUT:	Indicator of the existence of a palette

Purpose Opens the file, finds the next image, retrieves the dimensions of the image, and determines whether there is a palette associated with the image.

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

Description **DFR8getdims** retrieves the dimensions of the image and indicates whether a palette is associated and stored with the image. If the file is being opened for the first time, **DFR8getdims** returns information about the first image in the file. If an image has already been read, **DFR8getdims** finds the next image. Thus, images are read in the same order in which they were written to the file.

Normally, **DFR8getdims** is called before **DFR8getimage** so that if necessary, space allocations for the image and palette can be checked, and the dimensions can be verified. If this information is already known, **DFR8getdims** need not be called.

Valid values of *ispalette* are: `1` if there is a palette, or `0` if not.

FORTRAN

```
integer function d8gdims(filename, width, height, ispalette)

character* (*) filename
integer width, height
integer ispalette
```

DFR8getimage/d8gimg

intn DFR8getimage(char **filename*, uint8 **image*, int32 *width*, int32 *height*, uint8 **palette*)

filename	IN:	Name of the file
image	OUT:	Buffer for the returned image
width	IN:	Width of the image data buffer
height	IN:	Height of the image data buffer
palette	OUT:	Palette data

Purpose To retrieve the image and its palette, if it is present, and store them in the specified arrays.

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

Description In C, if *palette* is `NULL`, no palette is loaded, even if one is stored with the image. In Fortran-77, an array must be allocated to store the palette, even if no palette is expected to be stored. If the image in the file is compressed, **DFR8getimage** automatically decompresses it. If **DFR8getdims** has not been called, **DFR8getimage** finds the next image in the same way that **DFR8getdims** does.

The *width* and *height* parameters specify the number of columns and rows, respectively, in the array which you've allocated in memory to store the image. The image may be smaller than the allocated space.

The order in which you declare dimensions is different between C and Fortran-77. Ordering varies because Fortran-77 arrays are stored in column-major order, while C arrays are stored in row-major order. (Row-major order implies that the horizontal coordinate varies fastest). When **d8gimg** reads an image from a file, it assumes row-major order. The Fortran-77 declaration that causes an image to be stored in this way must have the width as its first dimension and the height as its second dimension. To take this into account as you read image in your program, the image must be built "on its side".

FORTRAN

```
integer function d8gimg(filename, image, width, height,
    palette)

character* (*) filename, image, palette
integer width, height
```

DFR8getpalref

intn DFR8getpalref(uint16 **pal_ref*)

pal_ref	OUT: Reference number of the palette

Purpose Retrieves the reference number of the palette associated with the last image accessed.

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

Description Make certain that **DFR8getdims** is called before **DFR8getpalref**.

DFR8lastref/d8lref

uint16 DFR8lastref( )

Purpose Retrieves the last reference number written to or read froman RIS8.

Return value Returns a non-zero reference number if successful and `FAIL` (or `-1`) otherwise.

Description This routine is primarily used for attaching annotations to images and adding images to vgroups. **DFR8lastref** returns the reference number of last raster image set read or written.

FORTRAN `integer function d8lref( )`

DFR8nimages/d8nims

intn DFR8nimages(char **filename*)

filename	IN:	Name of the HDF file

Purpose Retrieves the number of 8-bit raster images stored in the specified file.

Return value Returns the number of raster images in the file if successful and `FAIL` (or `-1`) otherwise.

FORTRAN

```
integer function d8nims(filename)

character* (*) filename
```

DFR8putimage/d8pimg

intn DFR8putimage(char **filename*, VOIDP *image*, int32 *width*, int32 *height*, uint16 *compress*)

filename	IN:	Name of the file to store the raster image in
image	IN:	Array with image to put in file
width	IN:	Number of columns in the image
height	IN:	Number of rows in the image
compress	IN:	Type of compression used, if any

Purpose Writes the RIS8 for the image as the first image in the file, overwriting any information previously in the file.

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

Description The *compress* parameter identifies the method to be used for compressing the data, if any. If IMCOMP compression is used, the image must include a palette.

DFR8putimage overwrites any information that exists in the HDF file. To write an image to a file by appending it, rather than overwriting it, use **DFR8addimage**.

In Fortran-77, the dimensions of the *image* array must be the same as the dimensions of the image itself.

The order in which dimensions are declared is different between C and Fortran-77. Ordering varies because Fortran-77 arrays are stored in column-major order, while C arrays are stored in row-major order. (Row-major order implies that the horizontal coordinate varies fastest). When **DFR8putimage** writes an image to a file, it assumes row-major order. The Fortran-77 declaration that causes an image to be stored in this way must have the width as its first dimension and the height as its second dimension, the reverse of the way it is done in C. To take this into account as you build your image in your Fortran-77 program, the image must be built "on its side".

FORTRAN

```
integer function d8pimg(filename, image, width, height,
                        compress)

character* (*) filename, image
integer width, height, compress
```

DFR8readref/d8rref

intn DFR8readref(char *filename, uint16 *ref*)

filename	IN:	Name of the file
ref	IN:	Reference number for next **DFR8getimage**

Purpose Specifies the reference number of the image to be read when **DFR8getimage** is next called.

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

Description **DFR8readref** is usually used in conjunction with **DFANlablist**, which returns a list of labels for a given tag together with their reference numbers. It provides, in a sense, a random access to images. There is no guarantee that reference numbers appear in sequence in an HDF file; therefore, it is not safe to assume that a reference number is the index of an image.

FORTRAN

```
integer function d8rref(filename, ref)

character* (*) filename
integer ref
```

DFR8restart/d8first

intn DFR8restart( )

Purpose	**DFR8restart** causes the next get command to read from the first raster image set in the file.
Return value	Returns `SUCCEED` (or `0`) if successful and `FAIL` (or `-1`) otherwise.

FORTRAN `integer function d8first( )`

DFR8setcompress/d8scomp

intn DFR8setcompress(int32 *type*, comp_info **cinfo*)

type	IN:	Type of compression
cinfo	IN:	Pointer to compression information structure

Purpose Sets the compression type to be used when writing the next 8-bit raster image.

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

Description This routine provides a method for compressing the next raster image written. The type can be one of the following values: `COMP_NONE`, `COMP_JPEG`, `COMP_RLE`, `COMP_IMCOMP`. `COMP_NONE` is the default for storing images if this routine is not called, therefore images are not compressed by default. `COMP_JPEG` compresses images with a JPEG algorithm, which is a lossy method. `COMP_RLE` uses lossless run-length encoding to store the image. `COMP_IMCOMP` uses a lossy compression algorithm called IMCOMP, and is included for backward compatibility only.

The `comp_info` union contains algorithm-specific information for the library routines that perform the compression and is defined in the `hcomp.h` header file as follows (refer to the header file for inline documentation):

```
typedef union tag_comp_info
  {
      struct
       {
            intn    quality;
            intn    force_baseline;
        }
      jpeg;
      struct
        {
            int32   nt;
            intn    sign_ext;
            intn    fill_one;
            intn    start_bit;
            intn    bit_len;
        }
      nbit;
      struct
        {
            intn    skp_size;
        }
      skphuff;
      struct
        {
            intn    level;
        }
      deflate;
  }
```

`comp_info;`

This union is defined to provide future expansion, but is currently only used by the `COMP_JPEG` compression type. A pointer to a valid `comp_info` union is required for all compression types other than `COMP_JPEG`, but the values in the union are not used. The `comp_info` union is declared in the header file hdf.h and is shown here for informative purposes only, it should not be re-declared in a user program.

For `COMP_JPEG` compression, the quality member of the jpeg structure must be set to the quality of the stored image. This number can vary from `100`, the best quality, to `0`, terrible quality. All images stored with `COMP_JPEG` compression are stored in a lossy manner, even images stored with a quality of 100. The ratio of size to perceived image quality varies from image to image, some experimentation may be required to determine an acceptable quality factor for a given application. The force_baseline parameter determines whether the quantization tables used during compression are forced to the range `0-255`. It should normally be set to `1` (forcing baseline results), unless special applications require non-baseline images to be used.

If the compression type is JPEG, **d8scomp** defines the default JPEG compression parameters to be used. If these parameters must be changed later, the **d8sjpeg** routine must be used. (Refer to the Reference Manual page on **d8sjpeg**).

FORTRAN

```
integer function d8scomp(type)

integer type
```

DFR8setpalette/d8spal

intn DFR8setpalette(uint8 **palette*)

palette	IN:	Palette data

Purpose Indicate which palette, if any, is to be used for subsequent image sets.

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

Description The specified palette remains the default palette until changed by a subsequent call to **DFR8setpalette**.

FORTRAN

```
integer function d8spal(palette)

character* (*) palette
```

DFR8writeref/d8wref

intn DFR8writeref(char **filename*, uint16 *ref*)

filename	IN:	Name of the HDF file
ref	IN:	Reference number for next call to **DFR8putimage** or **DFR8addimage**

Purpose Specifies the reference number of the image to be written when **DFR8addimage** or **DFR8putimage** is next called.

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

Description It is unlikely that you will need this routine, but if you do, use it with caution. There is no guarantee that reference numbers appear in sequence in an HDF file; therefore, it is not safe to assume that a reference number is the index of an image. In addition, using an existing reference number will overwrite the existing 8-bit raster image data.

FORTRAN

```
integer function d8wref(filename, ref)

character* (*) filename
integer ref
```