Chapter 7

24-Bit Raster Images (DF24 API)

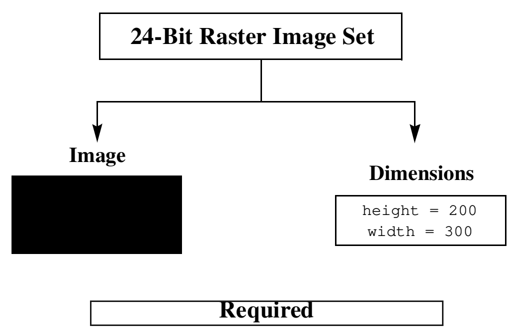
7.1 Chapter Overview

This chapter describes the 24-bit raster data model and the single-file DF24 routines available for storing and retrieving 24-bit raster images.

7.2 The 24-Bit Raster Data Model

The ***24-bit raster image set***, or ***RIS24***, data model supports two primary data objects: two-dimensional 24-bit raster images and dimensions. The primary member of the set is the ***24-bit raster image***, a two-dimensional array of pixels or picture elements. Each pixel is represented by three 8-bit numbers of image data. An optional compression method describes the method used, if any, to compress the image. Figure 7a shows the contents of a 24-bit raster image set.

FIGURE 7a **24-Bit Raster Image Set Contents**



7.2.1 Required 24-Bit Raster Image Data Set Objects

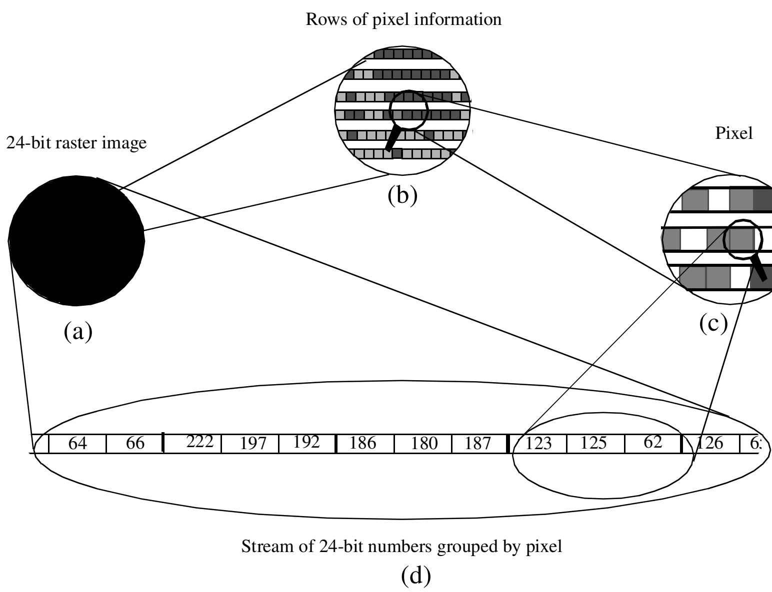
All 24-bit raster images must contain image data and a dimension record. These objects are created by the HDF library using information provided at the time the image is written to file.

7.2.1.1 24-Bit Raster Image Data Representation

The 24-bit raster image object contains a set of 24-bit pixel values, each of which has three 8-bit components; one for the red, one for the green, and one for the blue color component of the image. These values, referred to as ***RGB values***, are arranged in one of three specific ways, as described in Section 7.2.2.2 on page 187. The pixel values are arranged in rows, painted from left-to-right, top-to-bottom. As each pixel in a 24-bit image is represented in the image data by three 8-bit numbers, palettes are unnecessary and are not included in the 24-bit raster data model.

As an example, consider a stream of 24-bit numbers representing a raster image (Fig. 4.1a). To display the image, the color associated with the first number in the data stream appears in the upper left corner of the image. The remainder of the first line is then painted from left-to-right using as many values from the data stream as necessary to complete the line. The remainder of the rows are similarly painted from left-to-right, top-to-bottom until every value in the data stream appears as one pixel in the image.

FIGURE 7b **The Numerical Representation of a 24-Bit Raster Image**



A 24-bit raster image (a) is a set of rows displayed from left-to-right consisting of rows of pixels (b) whose values are stored as three 8-bit numbers (c) in a stream of data (d). In this figure, the image is interleaved by pixel.

7.2.1.2 24-Bit Raster Image Dimension

The ***dimensions*** of an image are the height and width of the image in pixels.

7.2.2 Optional 24-Bit Raster Image Data Set Objects

7.2.2.1 Compression Method

The only 24-bit compression method currently available in HDF is the JPEG algorithm . The applicable HDF compression tags are `COMP_JPEG`, and `COMP_NONE`. (See Table 7A.) The HDF tags `COMP_JPEG and COMP_NONE are` defined as the values `2`, and `0` respectively in the "hdf.h" header file.

TABLE 7A

24-Bit Raster Image Compression Method List

Compression Method	Type	Compression Code	Requirements
None	Lossless	COMP_NONE	Image data only (default setting).
JPEG	Lossy	COMP_JPEG	Image data, quality factor and compatibility factor.

JPEG Compression

The ***JPEG*** compression method is a lossy compression algorithm whose use is recommended for photographic or scanned images. Using JPEG compression to reduce the size of an image changes the values of the pixels and therefore may alter the meaning of the corresponding data.

For more information on the JPEG algorithm, refer to Chapter 6, titled *8-bit Raster Images (DFR8 API)*.

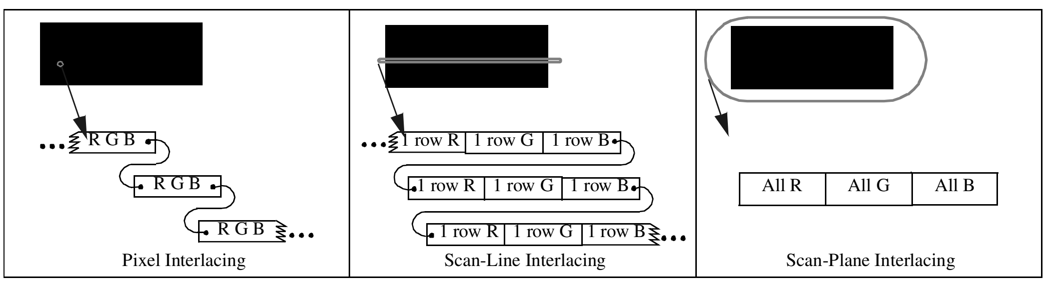
7.2.2.2 Interlace Modes

Because graphics applications and hardware devices vary in the way they access image data, HDF supports three interlace formats. By storing an image using a format that is consistent with the expected application or device, it is possible to achieve substantial improvements in performance.

HDF provides three options for organizing the color components in 24-bit raster images. These options consist of pixel interlacing, scan-line interlacing, and scan-plane interlacing. (See Figure 7c.) Storing the color components grouped by pixel, as in red-green-blue, red-green-blue, etc., is called ***pixel interlacing***. Storing the color components by line, as in one row of red, one row of green, one row of blue, one row red, etc., is called ***scan-line interlacing***. Finally, storing the color components grouped by color plane, as in the red components first, the green components second, and the blue components last, is called ***scan-plane interlacing***. Unless otherwise specified, the HDF 24-bit raster model assumes that all 24-bit images are stored using pixel interlacing.

FIGURE 7c

RGB Interlace Format for 24-Bit Raster Images



An interlace format describes both the physical format of an image as it is stored in memory and in the file. When writing to a file, HDF stores a 24-bit image using the same interlace format as it has in memory. However, when reading from a file, it is possible to make the in-core interlacing mode different from that used in the file. The following table contains a summary of the interlacing format available in the DF24 interface.

TABLE 7B

24-Bit Raster Image Interlace Format

format	Description	DF24setil or d2setil Parameter	Size of Image Array
Pixel	Components grouped by pixel.	DFIL_PIXEL	Width x Height x 3

Scan-line	Components grouped by row.	DFIL_LINE	Width x 3 x Height
Scan-plane	Components grouped by plane.	DFIL_PLANE	3 x Width x Height

7.3 The 24-Bit Raster Interface

The HDF library currently contains several routines for storing 24-bit raster images in the HDF format. The **DF24addimage**, **DF24putimage**, and **DF24getimage** routines are sufficient for most reading and writing operations.

7.3.1 24-Bit Raster Image Library Routines

The names of all C routines in the 24-bit raster image interface are prefaced by "DF24". The equivalent Fortran-77 routines are prefaced by "d2". These routines are divided into the following categories:

- ***Write routines*** create raster image sets and store them in new files or append them to existing files.
- ***Read routines*** determine the dimensions and interlace format of an image set, read the actual image data, and provide sequential or random read access to any raster image set.

The DF24 function calls are more explicitly defined in Table 7C and on their respective pages in the *HDF Reference Guide*.

TABLE 7C **DF24 Library Routines**

Purpose	Routine Name		Description
	C	Fortran-77	
Write	**DF24addimage**	**d2aimg**	Appends a 24-bit raster image to a file.
	DF24putimage	**d2pimg**	Writes a 24-bit raster image to file by overwriting all existing data.
	DF24setcompress	**d2scomp**	Sets the compression method for the next raster image written to the file.
	DF24setdims	**d2sdims**	Sets the dimensions for the next raster image written to the file.
	DF24setil	**d2setil**	Sets the interlace format of the next raster image written to the file.
	DF24lastref	**d2lref**	Reports the last reference number assigned to a 24-bit raster image.
	None	**d2sjpeg**	Fortran-specific routine for setting the parameters needed for the JPEG compression algorithm.
Read	**DF24getdims**	**d2gdims**	Retrieves the dimensions before reading the next raster image.
	DF24getimage	**d2gimg**	Reads the next 24-bit raster image.
	DF24reqil	**d2reqil**	Retrieves the interlace format before reading the next raster image.
	DF24readref	**d2rref**	Reads 24-bit raster image with the specified reference number.
	DF24restart	**d2first**	Returns to the first 24-bit raster image in the file.
	DF24nimage	**d2nimg**	Reports the number of 24-bit raster images in a file.

7.4 Writing 24-Bit Raster Images

The DF24 programming model for writing a 24-bit raster image set is as follows:

1. Set the interlace format if the interlacing is to be different from pixel interlacing. (optional)
2. Set the compression type if the image is to be compressed. (optional)
3. Write the raster data to the file.

Steps 1 and 2 can be invoked in any order, as long as they are executed before Step 3. By default, images are stored uncompressed using pixel interlacing.

7.4.1 Writing a 24-Bit Raster Image: DF24putimage and DF24addimage

To write a raster image to an HDF file, the calling program must contain one of the following function calls:

C: `status = DF24putimage(filename, image, width, height);`

FORTRAN: `status = d2pimg(filename, image, width, height)`

OR

C: `status = DF24addimage(filename, image, width, height);`

FORTRAN: `status = d2aimg(filename, image, width, height)`

DF24putimage and **DF24addimage** write a 24-bit raster images to the HDF file specified by the `filename` parameter. When given a new file name, **DF24putimage** and **DF24addimage** create a new file and write the raster image as the first raster image in the file. If a file with the specified filename exists, **DF24putimage** overwrites the previous contents of the file whereas **DF24addimage** appends data to the end of the file.

DF24putimage and **DF24addimage** passes the raster data in the `image` parameter and the width and height of the image in the `width` and `height` parameters. The array `image` is assumed to be the width times the height times three bytes in length for each color component. The parameters for **DF24putimage** and **DF24addimage** are further defined below. (See Table 7D on page 189.)

TABLE 7D

DF24putimage and DF24addimage Parameter List

Routine Name (Fortran-77)	Parameter	Data Type		Description
		C	Fortran-77	
DF24putimage (d2pimg)	filename	char *	character* (*)	Name of file to store the raster image.
	image	VOIDP	<valid numeric data type>	Raster image to be written.
	width	int32	integer	Number of columns in the image.
	height	int32	integer	Number of rows in the image.
DF24addimage (d2aimg)	filename	char *	character* (*)	Name of file to store the raster image.
	image	VOIDP	<valid numeric data type>	Raster image to be written.
	width	int32	integer	Number of columns in the image.
	height	int32	integer	Number of rows in the image.

EXAMPLE 1.

Writing a 24-Bit Raster Image to an HDF File

In the following examples, **DF24addimage** and **d2aimg** are used to write a 24-bit image to an HDF file named "Example1.hdf." **DF24addimage** assumes row-major order. Therefore, the Fortran-77 declaration requires the width (rows) before the height (columns), whereas the C declaration requires the height before the width. The interlace format setting is the default (pixel interlacing).

C:

```
#include "hdf.h"

#define WIDTH 5
#define HEIGHT 6
#define PIXEL_DEPTH 3

main( )
{

/* Initialize the image array. */
static uint8 raster_data[HEIGHT][WIDTH][PIXEL_DEPTH] =
{  1, 2, 3,  4, 5, 6,  7, 8, 9,  10,11,12,  13,14,15,
  16,17,18, 19,20,21, 22,23,24,  25,26,27,  28,29,30,
  31,32,33, 34,35,36, 37,38,39,  40,41,42,  43,44,45,
  46,47,48, 49,50,51, 52,53,54,  55,56,57,  58,59,60,
  61,62,63, 64,65,66, 67,68,69,  70,71,72,  73,74,75,
  76,77,78, 79,80,81, 82,83,84,  85,86,87,  88,89,90 };
    intn status;

/* Write the 24-bit raster image to the HDF file. */
status = DF24addimage("Example1.hdf", (VOIDP)raster_data, WIDTH, \
                  HEIGHT);

}
```

FORTRAN:

```
     PROGRAM WRITE RIS24

      integer status, d2aimg
      integer*4 WIDTH, HEIGHT, PIXEL_DEPTH
      parameter (WIDTH = 5,
     +         HEIGHT = 6,
     +         PIXEL_DEPTH = 3)

      character raster_data(PIXEL_DEPTH, WIDTH, HEIGHT)

C     Initialize the image array.
      data raster_data
     +     /  1, 2, 3,  4, 5, 6,  7, 8, 9,  10,11,12,  13,14,15,
     +     16,17,18, 19,20,21, 22,23,24,  25,26,27,  28,29,30,
     +     31,32,33, 34,35,36, 37,38,39,  40,41,42,  43,44,45,
     +     46,47,48, 49,50,51, 52,53,54,  55,56,57,  58,59,60,
     +     61,62,63, 64,65,66, 67,68,69,  70,71,72,  73,74,75,
     +     76,77,78, 79,80,81, 82,83,84,  85,86,87,  88,89,90 /

C     Write the 24-bit raster image to the file.
      status = d2aimg('Example1.hdf', raster_data, WIDTH,
     +                HEIGHT)

      end
```

7.4.2 Setting the Interlace Format: DF24setil

DF24setil indicates the interlace format to be used for all subsequent write operations. **DF24setil** changes the default setting from pixel interlacing to the selected format. When the format is set, it acts as the default until it is reset by another call to **DF24setil**. To change the default interlace format , the calling program must contain the following routines:

C:
```
status = DF24setil(il);
status = DF24addimage(filename, image, width, height);
```

FORTRAN:
```
status = d2setil(il)
status = d2aimg(filename, image, width, height)
```

DF24setil takes `il` as its only parameter. Valid values for `il` are `DFIL_PIXEL`, `DFIL_LINE`, and `DFIL_PLANE`. The parameters for **DF24setil** are further defined below. (See Table 7E on page 192.)

EXAMPLE 2.

Writing 24-Bit Raster Images Using Scan-plane Interlacing

In the following examples, **DF24addimage** is used to write a 24-bit image to an HDF file called "Example2.hdf". The **DF24setil** function used here to change the default format setting from pixel interlacing to scan-plane interlacing.

C:

```
#include "hdf.h"
#include "hcomp.h"

#define WIDTH 5
#define HEIGHT 6
#define PIXEL_DEPTH 3

main( )
{

/* Initialize the image array. */
static uint8 raster_data[HEIGHT][WIDTH][PIXEL_DEPTH] =
{  1, 2, 3,  4, 5, 6,  7, 8, 9,  10,11,12,  13,14,15,
16,17,18, 19,20,21, 22,23,24,  25,26,27,  28,29,30,
31,32,33, 34,35,36, 37,38,39,  40,41,42,  43,44,45,
46,47,48, 49,50,51, 52,53,54,  55,56,57,  58,59,60,
61,62,63, 64,65,66, 67,68,69,  70,71,72,  73,74,75,
76,77,78, 79,80,81, 82,83,84,  85,86,87,  88,89,90 };
intn status;

/* Change interlace from pixel to scan-plane. */
status = DF24setil(DFIL_PLANE);

/* Write the 24-bit image data to file. */
status = DF24addimage("Example2.hdf", (VOIDP)raster_data,
                WIDTH, HEIGHT);

}
```

FORTRAN:

```
     PROGRAM CHANGE INTERLACE

      integer status, d2aimg, d2setil
      integer*4 WIDTH, HEIGHT, PIXEL_DEPTH, DFIL_PLANE
      parameter (WIDTH = 5,
     +    HEIGHT = 6,
     +    PIXEL_DEPTH = 3,
     +    DFIL_PLANE = 2)

      integer raster_data(PIXEL_DEPTH, WIDTH, HEIGHT)

C     Initialize the image array.
```

```
      data raster_data
     + /  1, 2, 3,  4, 5, 6,  7, 8, 9,  10,11,12,  13,14,15,
     +  16,17,18, 19,20,21, 22,23,24,  25,26,27,  28,29,30,
     +  31,32,33, 34,35,36, 37,38,39,  40,41,42,  43,44,45,
     +  46,47,48, 49,50,51, 52,53,54,  55,56,57,  58,59,60,
     +  61,62,63, 64,65,66, 67,68,69,  70,71,72,  73,74,75,
     +  76,77,78, 79,80,81, 82,83,84,  85,86,87,  88,89,90  /

C     Change interlace from pixel to scan plane.
      status = d2setil(DFIL_PLANE)

C     Write the 24-bit raster image to the file.
      status = d2aimg('Example2.hdf', raster_data, WIDTH,
     +                 HEIGHT)

      end
```

7.4.3 Compressing Image Data: DF24setcompress and d2sjpeg

DF24setcompress invokes JPEG compression and sets the JPEG quality and baseline options. To store a 24-bit raster image using JPEG compression, the calling program must contain the following function calls:

C:
```
status = DF24setcompress(type, c_info);
status = DF24addimage(filename, image, width, height);
```

FORTRAN:
```
status = d2scomp(type) OR status = d2sjpeg(quality, baseline)
status = d2aimg(filename, image, width, height, compress)
```

Notice that the calling sequence for C is different from the calling sequence for Fortran-77. Once it is set, the parameter `type` in the **DF24setcompress** routine**,** or **d2scomp** in Fortran-77, routine specifies the compression method that will be used to store the raster images. However, the `c_info` parameter in **DF24setcompress** is missing from **d2scomp** which is a pointer to a structure that contains information specific to the compression method indicated by the `type` parameter. Because data structures of variable size are not supported in Fortran-77, a second compression-specific routine (**d2sjpeg**) is required in the Fortran-77 calling sequence.

For more information about the `c_info` structure refer to Chapter 6, titled *8-bit Raster Images (DFR8 API)*.

Default values for quality and baseline (`quality=75%`, `baseline=on`) are used if `c_info` is a null structure or **d2sjpeg** is omitted. Parameters for **DF24setcompress** and **d24sjpeg** are further described in Table 7E below.

TABLE 7E **DF24setil and DF24setcompress Parameter List**

Routine Name (Fortran-77)	Parameter	Data Type		Description
		C	Fortran-77	
DF24setil (d2sil)	il	int32	integer	Interlace format to be set.
DF24setcompress (d2scomp)	type	int32	integer	`COMP_JPEG`.
	c_info	comp_info *	None	Pointer to JPEG information structure.
(d2sjpeg)	quality	None	integer	JPEG compression quality specification.
	baseline	None	integer	JPEG compression baseline specification.

EXAMPLE 3.

Compressing and Writing a 24-Bit Raster Image

In the following examples, **DF24addimage** and **DF24compress** are used to compress a 24-bit image and write it to an HDF file named "Example2.hdf". Notice that compressing an image in C requires only one function call, whereas compressing an image using Fortran-77 requires two. The second Fortran-77 call is required because it is not valid to pass a structure as a parameter in Fortran-77.

C:

```
#include "hdf.h"
#include "hcomp.h"

#define WIDTH 3
#define HEIGHT 5
#define PIXEL_DEPTH 3

main( )
{
/* Initialize the image array. */
static uint8 raster_data[HEIGHT][WIDTH][PIXEL_DEPTH] =
    {  1, 2, 3,  4, 5, 6,  7, 8, 9,
      10,11,12, 13,14,15, 16,17,18,
      19,20,21, 22,23,24, 25,26,27,
      28,29,30, 31,32,33, 34,35,36,
      37,38,39, 40,41,42, 43,44,45 };
static comp_info compress_info;
intn status;

/* Initialize JPEG compression structure. */
compress_info.jpeg.quality = 60;
compress_info.jpeg.force_baseline = 1;

/* Set JPEG compression for storing the image. */
status = DF24setcompress(COMP_JPEG, &compress_info);

/* Write the 24-bit image data to file. */
status = DF24addimage("Example2.hdf", (VOIDP)raster_data,
                WIDTH, HEIGHT);

}
```

FORTRAN:

```
     PROGRAM COMPRESS RIS24

      integer d2aimg, d2scomp, d2sjpeg, status
      integer*4 WIDTH, HEIGHT, PIXEL_DEPTH
      parameter(WIDTH = 3,
     +          HEIGHT = 5,
     +          PIXEL_DEPTH = 3)
      character raster_data(PIXEL_DEPTH, WIDTH, HEIGHT)

C     Initialize the image array.
      data raster_data
     + /  1, 2, 3,  4, 5, 6,  7, 8, 9,
     +   10,11,12, 13,14,15, 16,17,18,
     +   19,20,21, 22,23,24, 25,26,27,
     +   28,29,30, 31,32,33, 34,35,36,
     +   37,38,39, 40,41,42, 43,44,45  /

C     Set compression.
      status = d2scomp(COMP_JPEG)
```

```
C     Set JPEG parameters to quality = 60, and turn compatibility on.
      status = d2sjpeg(60, 1)

C     Write the 24-bit image data to the HDF file.
      status = d2aimg('Example2.hdf', raster_data, WIDTH, HEIGHT)
      end
```

7.5 Reading 24-Bit Raster Images

The DF24 programming model for reading a 24-bit raster image set is as follows:

1. Determine the dimensions for an image if necessary.
2. Specify the interlace format to use when reading the image. (optional)
3. Read the image data from the file.

7.5.1 Reading a Raster Image: DF24getimage

If the dimensions and interlace format of the image are known, **DF24getimage** is the only function call required to read a raster image. If a file is being opened for the first time, **DF24getimage** returns the first image in the file. Additional calls will return successive images in the file, therefore images are read in the same order in which they were written to the file. Normally, **DF24getdims** and **DF24getil** are called before **DF24getimage** so that, if necessary, space allocations and interlace format for the image can be checked and the dimensions verified. If this information is already known, both function calls may be omitted.

The syntax of the **DF24getimage** routine is as follows:

C:
```
status = DF24getimage(filename, image, width, height);
```

FORTRAN:
```
status = d2gimg(filename, image, width, height)
```

DF24getimage retrieves the next 24-bit image from the HDF file specified by the `filename` parameter. If the image is compressed, **DF24getimage** decompresses it and places it in memory at the location pointed to by the `image` parameter. **DF24getimage** assumes the data is stored using pixel interlacing. The space allocated to hold the image is specified by the `width` and `height` parameters and may be larger than the actual image.The parameters for **DF24getimage** are further defined below. (See Table 7F on page 195.)

7.5.2 Determining the Dimensions of an Image: DF24getdims

DF24getdims opens a named file, finds the next image or the first image if the file is being opened for the first time, retrieves the dimensions of the image, then determines the interlace format of the image. Images are read in the order they were written.

To determine the dimensions and interlace format for an image, the calling program must call the following routines:

C:
```
status = DF24getdims(filename, width, height, il);
status = DF24getimage(filename, image, width, height);
```

FORTRAN:
```
status = d2gdim(filename, width, height, il)
status = d2gimg(filename, image, width, height)
```

DF24getdims takes four parameters: `filename`, `width`, `height`, and `il`. It retrieves dimension and interlace format information of the next 24-bit image stored in the HDF file specified by the `filename` parameter. The width and height are returned in the space pointed to by the `width` and `height` parameters respectively. The `il` parameter is used to determine the interlace format. The parameters for **DF24getdims** are further defined below. (See Table 7F.)

7.5.3 Modifying the Interlacing of an Image: DF24reqil

DF24reqil specifies an interlace format to be used when reading a 24-bit image from a file into memory. Regardless of what interlace format is used to store the image, **DF24reqil** forces the image to be loaded into memory using the specified interlace format.

To set or reset the interlace format, the calling program should call the following routines:

C:
```
status = DF24reqil(il);
status = DF24getimage(filename, image, width, height);
```

FORTRAN:
```
status = d2reqil(il)
status = d2gimg(filename, image, width, height)
```

DF24reqil takes `il` as its only parameter. Valid `il` values are `DFIL_PIXEL`, `DFIL_LINE` and `DFIL_PLANE`. As a call to **DF24reqil** may require a substantial reordering of the data, a much slower I/O performance than would be achieved if the interlace format wasn't reset may result.

The parameters of **DF24reqil** is further defined below. (See Table 7F.)

TABLE 7F

DF24getimage, DF24getdims and DF24reqil Parameter List

Routine Name (Fortran-77)	Parameter	Data Type		Description
		C	Fortran-77	
DF24getimage (d2gimg)	filename	char *	character* (*)	Name of the HDF file containing the raster image.
	image	VOIDP	<valid numeric data type>	Buffer for the raster image.
	width	int32	integer	Width of the raster image buffer.
	height	int32	integer	Height of the raster image buffer.
DF24getdims (d2gdims)	filename	char *	character* (*)	Name of HDF file containing the raster image.
	width	int32 *	integer	Pointer to the number of columns in the raster image.
	height	int32 *	integer	Pointer to the number of rows in the raster image.
	il	intn	integer	Pointer to the interlace format of the raster image.
DF24reqil (d2reqil)	il	intn	integer	Pointer to the interlace format of the raster image.

EXAMPLE 4.

Reading a 24-Bit Raster Image from an HDF File

The following examples read a 24-bit image from the "Example2.hdf" HDF file created in Example 2. Although the **DF24getdims** function call is optional, it is included as a demonstration of how to verify the image dimensions and interlace format before reading the image data. If the image dimensions and interlace format are known, only the **DF24getimage** call is required.

C:

```
#include "hdf.h"

#define WIDTH 5
#define HEIGHT 6
#define PIXEL_DEPTH 3

main( )
{
uint8 raster_data[PIXEL_DEPTH][HEIGHT][WIDTH];
int32 width, height;
intn interlace, status;

/* Get the image dimensions from the HDF file. */
status = DF24getdims("Example2.hdf", &width, &height,
                &interlace);

/*
* Read raster data if the dimensions are
* correct.
*/
if (width <= WIDTH && height <= HEIGHT)
  status = DF24getimage("Example2.hdf", (VOIDP)raster_data,
                width, height);
}
```

FORTRAN:

```
     PROGRAM READ RIS24

      integer d2gimg, d2gdims, status, width, height, interlace
      integer*4 X_LENGTH, Y_LENGTH, PIXEL_DEPTH
      parameter(X_LENGTH = 5, Y_LENGTH = 6, PIXEL_DEPTH = 3)
      integer raster_data(PIXEL_DEPTH, X_LENGTH, Y_LENGTH)

C     Read the dimensions raster image.
      status = d2gdims('Example2.hdf', width, height, interlace)

C     Read image data from the HDF file if the dimensions are
C     correct.
      if (width .eq. X_LENGTH .and. height .eq. Y_LENGTH) then
       status = d2gimg('Example2.hdf', raster_data, width, height)
      endif

      end
```

7.5.4 Reading a 24-Bit Raster Image with a Given Reference Number: DF24readref

DF24readref is used to access specific images stored in files containing multiple raster image sets. It is optionally used before **DF24getimage**. **DF24readref** can be used in connection with vgroups, which identify their members by tag/reference number pairs. See Chapter 5, titled *Vgroups (V API)*, for a discussion of vgroups and tag/reference number pairs.

To access a specific raster image set, use the following sequence of routine calls:

C:

```
status = DF24readref(filename, ref);
status DF24getimage(filename, image, width, height);
```

FORTRAN:

```
status = d2rref(filename, ref)
status = d2gimg(filename, image, width, height)
```

DF24readref sets the reference number for the next read operation performed on the HDF file `filename` to the reference number contained in `ref`. Because reference numbers are not always assigned in sequence, it is not guaranteed that a reference number represents the location of the image in the file.

The parameters of **DF24readref** are further described in the following table.

TABLE 7G **DF24readref Parameter List**

Routine Name (Fortran-77)	Parameter	Data Type		Description
		C	Fortran-77	
DF24readref (d2rref)	filename	char *	character* (*)	Name of HDF file containing the raster image.
	ref	uint16	integer	Reference number for the next call to **DF24getimage**.

7.5.5 Specifying that the Next Image Read to be the First 24-Bit Raster Image in the File: DF24restart

DF24restart causes the next call to **DF24getimage** or **DF24getdims** to read from the first raster image set in the file, rather than the RIS24 following the one that was most recently read. Use the following call to invoke **DF24restart**:

C: `status = DF24restart( );`

FORTRAN: `status = d2first( )`

TABLE 7H **DF24restart Parameter List**

Routine Name (Fortran-77)	Parameter	Data Type		Description
		C	Fortran-77	
DF24restart (d2first)	None	None	None	None.

7.6 24-Bit Raster Image Information Retrieval Routines

7.6.1 Querying the Total Number of Images in a File: DF24nimages

DF24nimages returns the total number of 24-bit raster image sets in a file, and has the following syntax:

C: `num_of_images = DF24nimages(filename);`

FORTRAN: `num_of_images = d2nimg(filename)`

TABLE 7I

DF24nimages Parameter List

Routine Name (Fortran-77)	Parameter	Data Type		Description
		C	Fortran-77	
DF24nimages (d2nimg)	filename	char *	character* (*)	Name of the HDF file.

7.6.2 Querying the Reference Number of the Most Recently Read or Written 24-Bit Raster Image: DF24lastref

DF24lastref returns the reference number of the 24-bit raster image most recently read or written. This routine is used for attaching annotations to images and adding images to vgroups. (See Chapter 5, titled *Vgroups (V API)* and Chapter 10, titled *Annotations (DFAN API)* for details on how to use reference numbers in connection with these applications.

The following calling sequence uses **DF24lastref** to find the reference number of the RIS24 most recently added to an HDF file:

C:

```
status = DF24addimage(filename, image, width, height, compress);
lastref = DF24lastref( );
```

FORTRAN:

```
status = d2aimg(filename, image, width, height, compress)
lastref = d2lref( )
```

DF24putimage or **DF24getimage** can be used in place of **DF24addimage** with similar results.

TABLE 7J

DF24lastref Parameter List

Routine Name (Fortran-77)	Parameter	Data Type		Description
		C	Fortran-77	
DF24lastref (d2lref)	filename	None	None	None.