
C:
int DFPrestart ()

Purpose: To cause the next call to DFPrestart to read the first palette in the file.

Returns: 0 on success.

DFPlastref

FORTTRAN:
INTEGER FUNCTION DFPlastref ()

C:
int DFPlastref ()

Purpose: To determine the value of the reference number most recently read or written by a palette function call.

Returns: The reference number on success; -1 on failure.

Other Palette Routines

DFPnpals

FORTTRAN:
 INTEGER FUNCTION DFPnpals(filename)
 CHARACTER*(*) filename

C:
 int DFPnpals(filename)
 char *filename; /* name of HDF file */

Purpose: To tell how many palettes are present in a file.

Returns: Number of palettes on success; -1 on failure.

DFPwriteref

FORTTRAN:
 INTEGER FUNCTION DFPwriteref(filename, ref)
 CHARACTER*(*) filename - name of HDF file
 INTEGER ref - ref number to be used in next
 palette write

C:
 int DFPwriteref(filename, ref)
 char *filename; /* name of HDF file */
 uint16 ref; /* ref number to be used in next palette write */

Purpose: To set the reference number of the next palette to be written.

Returns: 0 on success; -1 on failure.

DFPreadref

FORTTRAN:
 INTEGER FUNCTION DFPreadref(filename, ref)
 CHARACTER*(*) filename - name of HDF file
 INTEGER ref - ref number to be used in next read

C:
 int DFPreadref(filename, ref)
 char *filename; /* name of HDF file */
 uint16 ref; /* ref number to be used in next DFPgetpal */

Purpose: To set the reference number of the palette that DFPgetpal will retrieve next.

Returns: 0 on success; -1 if a palette with this reference number does not exist or if an error occurs.

DFPrestart

FORTTRAN:
 INTEGER FUNCTION DFPrestart()

Purpose: To write a palette to file.

Returns: 0 on success; -1 on failure.

This routine provides more control than `DFPaddpal`. Note that the combination `filemode="w"` and `overwrite=1` has no meaning and will generate an error.

Reading Palettes from a File

DFPgetpal

FORTTRAN:

```
INTEGER FUNCTION DFPgetpal(filename, pal)
CHARACTER*(*) filename    - name of HDF file
CHARACTER*(*) pal         - 768-byte space for palette
```

C:

```
int DFPgetpal(filename, palette)
char *filename;           /* name of HDF file */
unsigned char *palette;   /* 768-byte space for palette */
```

Purpose: To get the next palette from file and store it in the array `palette`.

Returns: 0 on success; -1 on failure.

The array `palette` is assumed to be allocated at least 768 bytes. Successive additional calls to `DFPgetpal` retrieve the palettes in the file in the sequence in which they are stored.

`DFPgetpal` is often sufficient for getting the palette that you want from an HDF file. Other palette routines, which provide more refined access to palettes are described below.

Example: Reading the First Available Palette

Figure 3.2 shows a C code segment that reads the first available palette from an HDF file.

Figure 3.2 Reading the First Available Palette

```
C:
int          DFPgetpal();
char         pal[768];

DFPgetpal("myfile.hdf", pal);
...
```

Chapter Overview

This chapter describes the routines that are available for storing and retrieving 8-bit palettes.

HDF 8-Bit Palettes

An *8-bit palette* is a lookup table with 256 entries that tell the system hardware which color to associate with each of the 256 possible pixel values. Each entry in the palette is chosen from a master palette of 2^{24} RGB colors.

In HDF files, 8-bit palettes are assumed to be organized as follows. Each palette entry consists of 3 bytes—one each for red, green, and blue. The first three bytes represent the R, G, and B values of the first color in the palette; the next three the R, G, and B values of the second color; and so forth. The total size of a palette is 768 bytes.

The HDF library contains routines for storing and retrieving palettes. These routines are callable from C and FORTRAN programs that have access to the library. All of the callable palette routines in the library begin with the letters DFP.

The functions, `DFPaddpal` and `DFPgetpal`, are the primary routines for palette I/O and should be sufficient for most palette I/O operations. The other six palette functions—`DFPputpal`, `DFPnpals`, `DFPwriteref`, `DFPreadref`, `DFPrestart`, and `DFPlastref`—provide greater control of the I/O process and are available to you if more control is needed.

Table 2.1 lists the long and short names and the functions of the palette routines currently contained in the HDF library. The following sections provide descriptions and examples of these calling routines.

Table 3.1 Palette I/O Routines in the HDF Library

Long Name	Short Name	Function
<code>DFPaddpal</code>	<code>dpapal</code>	appends a palette to a file.
<code>DFPgetpal</code>	<code>dpgpal</code>	reads in the next palette in the file.
<code>DFPputpal</code>	<code>dpppal</code>	writes a palette to a file.
<code>DFPnpals</code>	<code>dpnpals</code>	indicates number of palettes in a file.
<code>DFPwriteref</code>	<code>dpwref</code>	sets the reference number of the next palette to be written.
<code>DFPreadref</code>	<code>dprref</code>	sets the reference number of the next palette to be retrieved.
<code>DFPrestart</code>	<code>dprest</code>	specifies that the next call to <code>DFPgetpal</code> reads the first palette in the file, rather than the next.
<code>DFPlastref</code>	<code>dplref</code>	returns the value of the reference number most recently read or written.

Chapter **3**

Storing Palettes

Chapter Overview

HDF 8-Bit Palettes

Writing Palettes to a File

Reading Palettes from a File

Other Palette Routines