

# DFPaddpal/dpapal

---

## DFPaddpal/dpapal

intn DFPaddpal(const char \**filename*, const VOIDP *palette*)

<i>filename</i>	IN: Name of the HDF file
<i>palette</i>	IN: Buffer containing the palette to be written

**Purpose** Appends a palette to a file.

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

**Description** If the named file does not exist, it is created and the palette written to it. The *palette* buffer should be at least 768 bytes in length.

FORTRAN

```
integer function dpapal(filename, palette)  
  
character* (*) filename, palette
```

**DFPgetpal/dpgpal**

intn DFPgetpal(const char \**filename*, VOIDP *palette*)

<i>filename</i>	IN: Name of the HDF file
<i>palette</i>	OUT: Buffer for the returned palette

<b>Purpose</b>	Retrieves the next palette from file and stores it in the buffer <i>palette</i> .
<b>Return value</b>	Returns <code>SUCCEED</code> (or 0) if successful and <code>FAIL</code> (or -1) otherwise.
<b>Description</b>	The <i>palette</i> buffer is assumed to be at least 768 bytes long. Successive calls to <b>DFPgetpal</b> retrieve the palettes in the sequence they are stored in the file.

FORTTRAN	<code>integer function dpgpal(filename, palette)</code>  <code>character* (*) filename. palette</code>
----------	--

# DFPlastref/dplref

---

## DFPlastref/dplref

uint16 DFPlastref( void )

**Purpose** Returns the value of the reference number most recently read or written by a palette function call.

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

**FORTRAN** `integer function dplref( )`

**DFPnpals/dpnpals**

intn DFPnpals(const char \**filename*)

*filename*            IN:    Name of the file

**Purpose**            Indicates the number of palettes in the specified file.

**Return value**      Returns the number of palettes if successful and `FAIL` (or `-1`) otherwise.

FORTTRAN            integer function dpnpals(filename)  
                      character\* (\*) filename

# DFPputpal/dpppal

---

## DFPputpal/dpppal

intn DFPputpal (const char \**filename*, const VOIDP *palette*, intn *overwrite*, const char \**filemode*)

<i>filename</i>	IN: Name of the file
<i>palette</i>	IN: Buffer containing the palette to be written
<i>overwrite</i>	IN: Flag identifying the palette to be written
<i>filemode</i>	IN: File access mode

**Purpose** Writes a palette to the file.

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

**Description** This routine provides more control of palette write operations than **DFPaddpal**. Note that the combination *filemode*="w" and *overwrite*=1 has no meaning and will result in an error condition. To overwrite a palette, *filename* must be the same filename as the last file accessed through the DFP interface.

Valid values for *overwrite* are: 1 to overwrite last palette; 0 to write a new palette.

Valid values for *filemode* are: "a" to append the palette to the file and "w" to create a new file.

The *palette* buffer must be at least 768 bytes in length.

FORTRAN

```
integer function dpppal(filename, palette, overwrite,  
                        filemode)  
  
character* (*) filename, palette, filemode  
integer overwrite
```

**DFPreadref/dprref**

intn DFPreadref(const char \**filename*, uint16 *ref*)

<i>filename</i>	IN: Name of the file
<i>ref</i>	IN: Reference number to be used in next <b>DFPgetpal</b> call

<b>Purpose</b>	Retrieves the reference number of the palette to be retrieved next by <b>DFPgetpal</b> .
----------------	--

<b>Return value</b>	Returns <code>SUCCESS</code> (or 0) if the palette with the specified reference number exists and <code>FAIL</code> (or -1) otherwise.
---------------------	--

<b>Description</b>	Used to set the reference number of the next palette to be retrieved.
--------------------	---

FORTTRAN	<pre>integer function dprref(filename, ref)  character* (*) filename integer ref</pre>
----------	--

# DFPrestart/dprest

---

## DFPrestart/dprest

intn DFPrestart( void )

**Purpose** Specifies that **DFPgetpal** will read the first palette in the file, rather than the next unread palette.

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

**FORTRAN** `integer function dprest( )`

**DFPwriteref/dpwref**

intn DFPwriteref(const char \**filename*, uint16 *ref*)

<i>filename</i>	IN:	Name of the file
<i>ref</i>	IN:	Reference number to be assigned to the next palette written to a file

**Purpose** Determines the reference number of the next palette to be written.

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

**Description** The file name is ignored. The next palette written, regardless of the filename, is assigned the reference number *ref*.

FORTTRAN

```
integer function dpwref(filename, ref)

character* (*) filename
integer ref
```