

A Portable Floating-Point-Based Image Format

1.0

Generated by Doxygen 1.7.6.1

Sat Feb 4 2012 19:46:37

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	pfi.c File Reference	3
2.1.1	Define Documentation	4
2.1.1.1	DPRINTF	4
2.1.1.2	PIXEL_PFIS	4
2.1.1.3	SIZE_CHECK	4
2.1.2	Function Documentation	4
2.1.2.1	double2pfi	4
2.1.2.2	numexpect	5
2.1.2.3	pfi2double	5
2.1.2.4	pfi_getv	6
2.1.2.5	pfi_properties	6
2.1.2.6	pfi_properties_name	7
2.1.2.7	pfi_putv	7
2.1.2.8	pfi_read	7
2.1.2.9	pfi_read_internal	8
2.1.2.10	pfi_read_name	8
2.1.2.11	pfi_write	9
2.1.2.12	pfi_write_name	10
2.1.2.13	SIZE_CHECK	10
2.1.2.14	SIZE_CHECK	10
2.1.2.15	skipcomments	10
2.1.2.16	skipwhite	11

2.2	pfi.h File Reference	11
2.2.1	Define Documentation	12
2.2.1.1	DLL_EXPORT	12
2.2.1.2	PFI_EPSILON	12
2.2.1.3	pfi_ftoh16	12
2.2.1.4	pfi_ftoh32	12
2.2.1.5	pfi_htof16	12
2.2.1.6	pfi_htof32	13
2.2.1.7	PFI_VERSION	13
2.2.2	Enumeration Type Documentation	13
2.2.2.1	anonymous enum	13
2.2.3	Function Documentation	13
2.2.3.1	pfi_getv	13
2.2.3.2	pfi_properties	14
2.2.3.3	pfi_properties_name	14
2.2.3.4	pfi_putv	15
2.2.3.5	pfi_read	15
2.2.3.6	pfi_read_name	15
2.2.3.7	pfi_write	16
2.2.3.8	pfi_write_name	17
2.3	pfi.h File Reference	17
2.3.1	Define Documentation	19
2.3.1.1	DLL_EXPORT	19
2.3.1.2	PFI_EPSILON	19
2.3.1.3	pfi_ftoh16	19
2.3.1.4	pfi_ftoh32	19
2.3.1.5	pfi_htof16	19
2.3.1.6	pfi_htof32	19
2.3.1.7	PFI_VERSION	19
2.3.2	Enumeration Type Documentation	19
2.3.2.1	anonymous enum	19
2.3.3	Function Documentation	20
2.3.3.1	pfi_getv	20
2.3.3.2	pfi_properties	20

2.3.3.3	pfi_properties_name	20
2.3.3.4	pfi_putv	21
2.3.3.5	pfi_read	21
2.3.3.6	pfi_read_name	22
2.3.3.7	pfi_write	22
2.3.3.8	pfi_write_name	23
2.4	sizecheck.c File Reference	24
2.4.1	Function Documentation	24
2.4.1.1	main	24

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

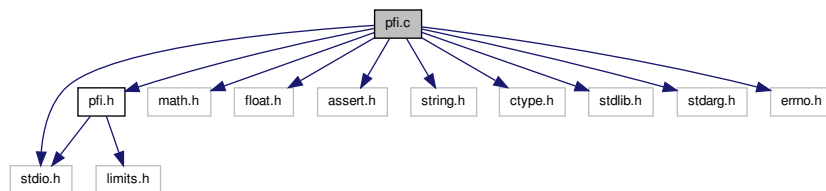
pfi.c	3
debian/tmp/usr/include/pfi.h	11
pfi.h	17
sizecheck.c	24

Chapter 2

File Documentation

2.1 pfi.c File Reference

```
#include "pfi.h" #include <stdio.h> #include <math.h> ×  
#include <float.h> #include <assert.h> #include <string.-  
h> #include <ctype.h> #include <stdlib.h> #include <stdarg.-  
h> #include <errno.h> Include dependency graph for pfi.c:
```



Defines

- #define [SIZE_CHECK](#)(type, size, name) typedef type name[(sizeof (type) == size)*2-1]
- #define [PIXEL_PFIS](#) 7
- #define [DPRINTF](#)(x)

Functions

- [SIZE_CHECK](#) (pfiu32, 4, bar)
- [SIZE_CHECK](#) (pfiu16, 2, foobar)
- static void [double2pfi](#) (double val, pfiu32 *results)

- static int [pfi2double](#) (pfiu32 *coded, double *res)
- static int [numexpect](#) (char *bufp, int expect)
- static void [skipwhite](#) (char **pos)
- static int [skipcomments](#) (FILE *stream)
- static int [pfi_read_internal](#) (FILE *, int, int, int, double *)
- int [pfi_getv](#) (FILE *stream, double *valp)
- int [pfi_putv](#) (FILE *stream, double val)
- int [pfi_properties](#) (FILE *stream, int *rows, int *cols, int *colors, int *colormodel)
- double * [pfi_read](#) (FILE *stream, int *rows, int *cols, int *colors, int *colormodel)
- int [pfi_write](#) (FILE *stream, double *buf, int rows, int cols, int colors, int colormodel, char *fname)
- [DLL_EXPORT](#) int [pfi_read_name](#) (char *name, int rows, int cols, int colors, double *result)
- [DLL_EXPORT](#) int [pfi_write_name](#) (char *name, double *buf, int rows, int cols, int colors, int colormodel)
- [DLL_EXPORT](#) int [pfi_properties_name](#) (char *name, int *rows, int *cols, int *colors, int *colormodel)

2.1.1 Define Documentation

2.1.1.1 #define **DPRINTF**(x)

Definition at line 81 of file pfi.c.

Referenced by [double2pfi\(\)](#), [pfi2double\(\)](#), [pfi_getv\(\)](#), [pfi_properties\(\)](#), [pfi_putv\(\)](#), [pfi_read\(\)](#), and [pfi_read_internal\(\)](#).

2.1.1.2 #define **PIXEL_PFIS** 7

Definition at line 79 of file pfi.c.

Referenced by [pfi_getv\(\)](#), [pfi_putv\(\)](#), and [pfi_read_internal\(\)](#).

2.1.1.3 #define **SIZE_CHECK**(type, size, name) typedef type name[(sizeof (type) == size)*2-1]

Definition at line 22 of file pfi.c.

2.1.2 Function Documentation

2.1.2.1 static void **double2pfi** (double val, pfiu32 * results) [static]

{enumerate}

The raw data consists of two 16 bit and six 32 bit unsigned integers in network byte ordering (byte-swapped on intel systems) per pixel color. We assume exponential notation, like with e in printf. The first two numbers contain the number left from the decimal

point and the exponent respectively. For negative values the two- complement form is used. Every of six 32-bit numbers contains 9 digits of the number right from the decimal point, ordered from left to right, so a precision of 54 digits can be achieved.

Calculates 7 pfiu32 values to represent one double.

Parameters

<i>val</i>	The double value.
<i>results</i>	A reference to an array of 7 pfiu32 that will hold the result.

Definition at line 1051 of file pfi.c.

References DPRINTF.

Referenced by pfi_putv().

2.1.2.2 static int **numexpect** (char * bufp, int expect) [static]

Checks a string for the correct format of a given number of numbers.

Parameters

<i>bufp</i>	The string that contains the number we want.
<i>expect</i>	the expected value

Returns

The expected number or -1 on error.

Definition at line 1184 of file pfi.c.

References skipwhite().

Referenced by pfi_properties().

2.1.2.3 static int **pfi2double** (pfiu32 * coded, double * res) [static]

Calculates a double from 7 pfiu32 values.

Parameters

<i>coded</i>	A reference to an array of four pfiu32.
<i>res</i>	A reference to the double representation of the four pfiu32.

Returns

0 on succes, -1 on error.

Definition at line 1125 of file pfi.c.

References DPRINTF.

Referenced by `pfi_getv()`.

2.1.2.4 `int pfi_getv (FILE * stream, double * valp)`

Reads a double value from an open pfi file

Parameters

<i>stream</i>	The open file.
<i>valp</i>	A reference to the value to be read.

Returns

0 on success, else -1.

Definition at line 380 of file `pfi.c`.

References `DPRINTF`, `pfi2double()`, `pfi_ftoh16`, `pfi_ftoh32`, and `PIXEL_PFIS`.

Referenced by `pfi_read_internal()`.

2.1.2.5 `int pfi_properties (FILE * stream, int * rows, int * cols, int * colors, int * colormodel)`

Gets the pfi image properties from an open file.

Parameters

<i>stream</i>	The open file.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	A reference to an integer to represent the colormodel used in this image to be determined from the image header. We use 0 for none (greyscale), 1 for RGB etc.

Returns

success: 0, else -1.

Definition at line 489 of file `pfi.c`.

References `DPRINTF`, `numexpect()`, and `skipcomments()`.

Referenced by `pfi_properties_name()`, and `pfi_read()`.

2.1.2.6 **DLL_EXPORT** int **pfi_properties_name** (char * name, int * rows, int * cols, int * colors, int * colormodel)

Gets the pfi image properties from an image file.

Parameters

<i>name</i>	The file name.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	A reference to an integer to represent the colormodel used in this image to be determined from the image header. We use 0 for none (greyscale), 1 for RGB etc.

Returns

success: 0, else -1.

Definition at line 888 of file pfi.c.

References `pfi_properties()`.

2.1.2.7 int **pfi_putv** (FILE * stream, double val)

Writes a double value into an open pfi file.

Parameters

<i>stream</i>	The open file.
<i>val</i>	The value to be written.

Returns

0 on success, else -1.

Definition at line 428 of file pfi.c.

References `double2pfi()`, `DPRINTF`, `pfi_htof16`, `pfi_htof32`, and `PIXEL_PFIS`.

Referenced by `pfi_write()`.

2.1.2.8 double* **pfi_read** (FILE * stream, int * rows, int * cols, int * colors, int * colormodel)

Reads a pfi image from an open file.

Parameters

<i>stream</i>	The open file.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	A reference to an integer to represent the colormodel used in this image to be determined from the image header. We use 0 for none (greyscale), 1 for RGB etc.

Returns

A new double array containing the image. The colors are stored one after another, so we've got pixel one with all colors first, then pixel two etc. In case of error, NULL is returned

Definition at line 654 of file pfi.c.

References `DPRINTF`, `pfi_properties()`, and `pfi_read_internal()`.

```
2.1.2.9 static int pfi_read_internal ( FILE * stream, int rows, int cols, int colors, double *
      res ) [static]
```

Reads a pfi image from an open file.

Parameters

<i>stream</i>	The open file.
<i>rows</i>	the number of rows
<i>cols</i>	the number of cols
<i>colors</i>	the number of colors
<i>res</i>	A double array containing the image. The colors are stored one after another, so we've got pixel one with all colors first, then pixel two etc.

Returns

0 if successful, else -1

Definition at line 982 of file pfi.c.

References `DPRINTF`, `pfi_getv()`, and `PIXEL_PFIS`.

Referenced by `pfi_read()`, and `pfi_read_name()`.

```
2.1.2.10 DLL_EXPORT int pfi_read_name ( char * name, int rows, int cols, int colors,
      double * result )
```

Reads a pfi image.

Parameters

<i>name</i>	The file name.
<i>rows</i>	the number of rows
<i>cols</i>	the number of cols
<i>colors</i>	the number of colors
<i>result</i>	A double array containing the image. The colors are stored one after another, so we've got pixel one with all colors first, then pixel two etc.

Returns

0 if successful, else -1

Definition at line 770 of file pfi.c.

References `pfi_read_internal()`.

2.1.2.11 int **pfi_write** (FILE * stream, double * buf, int rows, int cols, int colors, int colormodel, char * fname)

Writes a pfi image to an open file.

Parameters

<i>stream</i>	The open file.
<i>buf</i>	A reference to the buffer containing the image. The colors are stored one after another, so we've got pixel one with all colors first, then pixel two etc.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	An integer to represent the colormodel used in this image. We use 0 for none (greyscale), 1 for RGB etc.
<i>fname</i>	The name which will be written to the file header.

Returns

0 on success, else -1.

Definition at line 715 of file pfi.c.

References `pfi_putv()`.

Referenced by `pfi_write_name()`.

2.1.2.12 **DLL_EXPORT** int **pfi_write_name** (char * name, double * buf, int rows, int cols, int colors, int colormodel)

Opens and writes a pfi image.

Parameters

<i>name</i>	The file name.
<i>buf</i>	A reference to the buffer containing the image. The colors are stored one after another, so we've got pixel one with all colors first, then pixel two etc.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	An integer to represent the colormodel used in this image. We use 0 for none (greyscale), 1 for RGB etc.

Returns

0 on success, else -1.

Definition at line 833 of file pfi.c.

References pfi_write().

2.1.2.13 **SIZE_CHECK** (pfiu32 , 4 , bar)

2.1.2.14 **SIZE_CHECK** (pfiu16 , 2 , foobar)

2.1.2.15 static int **skipcomments** (FILE * stream) [static]

Moves the file position pointer to the next line that is no comment line.

Parameters

<i>stream</i>	The open file.
---------------	----------------

Returns

The new line's first character or EOF at EOF.

Definition at line 927 of file pfi.c.

Referenced by pfi_properties().

2.1.2.16 static void **skipwhite**(char** pos) [static]

Returns the position of the first non whitespace character in a string.

Parameters

<i>pos</i>	A string.
------------	-----------

Returns

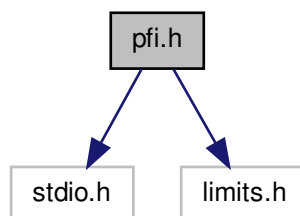
The new position.

Definition at line 1214 of file pfi.c.

Referenced by numexpect().

2.2 pfi.h File Reference

#include <stdio.h> #include <limits.h> Include dependency graph for debian/tmp/usr/include/pfi.h:



Defines

- #define `PFI_VERSION` "1.2"
- #define `PFI_EPSILON` (3 * DBL_EPSILON)
- #define `DLL_EXPORT`
- #define `pfi_htof32`(x) (x)
- #define `pfi_ftoh32`(x) (x)
- #define `pfi_htof16`(x) (x)
- #define `pfi_ftoh16`(x) (x)

Enumerations

- enum { `pfi_grey` = 0, `pfi_rgb`, `pfi_yuv`, `pfi_unknown` }

Functions

- int `pfi_getv` (FILE *stream, double *valp)
- int `pfi_putv` (FILE *stream, double val)
- int `pfi_properties` (FILE *stream, int *rows, int *cols, int *colors, int *colormodel)
- double * `pfi_read` (FILE *stream, int *rows, int *cols, int *colors, int *colormodel)
- int `pfi_write` (FILE *stream, double *buf, int rows, int cols, int colors, int colormodel, char *fname)
- `DLL_EXPORT` int `pfi_read_name` (char *name, int rows, int cols, int colors, double *result)
- `DLL_EXPORT` int `pfi_write_name` (char *name, double *buf, int rows, int cols, int colors, int colormodel)
- `DLL_EXPORT` int `pfi_properties_name` (char *name, int *rows, int *cols, int *colors, int *colormodel)

2.2.1 Define Documentation

2.2.1.1 #define `DLL_EXPORT`

Definition at line 36 of file `debian/tmp/usr/include/pfi.h`.

2.2.1.2 #define `PFI_EPSILON` (3 * DBL_EPSILON)

Definition at line 28 of file `debian/tmp/usr/include/pfi.h`.

2.2.1.3 #define `pfi_ftoh16`(x)(x)

Definition at line 131 of file `debian/tmp/usr/include/pfi.h`.

Referenced by `pfi_getv`().

2.2.1.4 #define `pfi_ftoh32`(x)(x)

Definition at line 129 of file `debian/tmp/usr/include/pfi.h`.

Referenced by `pfi_getv`().

2.2.1.5 #define `pfi_htof16`(x)(x)

Definition at line 130 of file `debian/tmp/usr/include/pfi.h`.

Referenced by `pfi_putv`().

2.2.1.6 #define **pfi_htof32**(x) (x)

Definition at line 128 of file debian/tmp/usr/include/pfi.h.

Referenced by pfi_putv().

2.2.1.7 #define **PFI_VERSION** "1.2"

Definition at line 26 of file debian/tmp/usr/include/pfi.h.

2.2.2 Enumeration Type Documentation

2.2.2.1 anonymous enum

Colormodel specifier. Either greyscale, rgb or yuv.

Enumerator:

pfi_grey

pfi_rgb

pfi_yuv

pfi_unknown

Definition at line 136 of file debian/tmp/usr/include/pfi.h.

2.2.3 Function Documentation

2.2.3.1 int **pfi_getv** (FILE * stream, double * valp)

Reads a double value from an open pfi file

Parameters

<i>stream</i>	The open file.
<i>valp</i>	A reference to the value to be read.

Returns

0 on success, else -1.

Definition at line 380 of file pfi.c.

Referenced by pfi_read_internal().

2.2.3.2 **int pfi_properties** (FILE * stream, int * rows, int * cols, int * colors, int * colormodel)

Gets the pfi image properties from an open file.

Parameters

<i>stream</i>	The open file.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	A reference to an integer to represent the colormodel used in this image to be determined from the image header. We use 0 for none (greyscale), 1 for RGB etc.

Returns

success: 0, else -1.

Definition at line 489 of file pfi.c.

Referenced by pfi_properties_name(), and pfi_read().

2.2.3.3 **DLL_EXPORT int pfi_properties_name** (char * name, int * rows, int * cols, int * colors, int * colormodel)

Gets the pfi image properties from an image file.

Parameters

<i>name</i>	The file name.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	A reference to an integer to represent the colormodel used in this image to be determined from the image header. We use 0 for none (greyscale), 1 for RGB etc.

Returns

success: 0, else -1.

Definition at line 888 of file pfi.c.

2.2.3.4 int **pfi_putv** (FILE * stream, double val)

Writes a double value into an open pfi file.

Parameters

<i>stream</i>	The open file.
<i>val</i>	The value to be written.

Returns

0 on success, else -1.

Definition at line 428 of file pfi.c.

Referenced by pfi_write().

2.2.3.5 double* **pfi_read** (FILE * stream, int * rows, int * cols, int * colors, int * colormodel)

Reads a pfi image from an open file.

Parameters

<i>stream</i>	The open file.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	A reference to an integer to represent the colormodel used in this image to be determined from the image header. We use 0 for none (greyscale), 1 for RGB etc.

Returns

A new double array containing the image. The colors are stored one after another, so we've got pixel one with all colors first, then pixel two etc. In case of error, NULL is returned

Definition at line 654 of file pfi.c.

2.2.3.6 **DLL_EXPORT** int **pfi_read_name** (char * name, int rows, int cols, int colors, double * result)

Reads a pfi image.

Parameters

<i>name</i>	The file name.
<i>rows</i>	the number of rows
<i>cols</i>	the number of cols
<i>colors</i>	the number of colors
<i>result</i>	A double array containing the image. The colors are stored one after another, so we've got pixel one with all colors first, then pixel two etc.

Returns

0 if successful, else -1

Definition at line 770 of file pfi.c.

2.2.3.7 int **pfi_write** (FILE * stream, double * buf, int rows, int cols, int colors, int colormodel, char * fname)

Writes a pfi image to an open file.

Parameters

<i>stream</i>	The open file.
<i>buf</i>	A reference to the buffer containing the image. The colors are stored one after another, so we've got pixel one with all colors first, then pixel two etc.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	An integer to represent the colormodel used in this image. We use 0 for none (greyscale), 1 for RGB etc.
<i>fname</i>	The name which will be written to the file header.

Returns

0 on success, else -1.

Definition at line 715 of file pfi.c.

Referenced by pfi_write_name().

2.2.3.8 **DLL_EXPORT** int **pfi_write_name** (char * name, double * buf, int rows, int cols, int colors, int colormodel)

Opens and writes a pfi image.

Parameters

<i>name</i>	The file name.
<i>buf</i>	A reference to the buffer containing the image. The colors are stored one after another, so we've got pixel one with all colors first, then pixel two etc.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	An integer to represent the colormodel used in this image. We use 0 for none (greyscale), 1 for RGB etc.

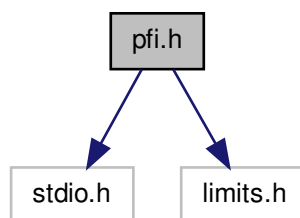
Returns

0 on success, else -1.

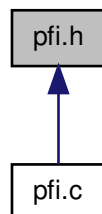
Definition at line 833 of file pfi.c.

2.3 pfi.h File Reference

#include <stdio.h> #include <limits.h> Include dependency graph for pfi.h:



This graph shows which files directly or indirectly include this file:



Defines

- `#define PFI_VERSION "1.2"`
- `#define PFI_EPSILON (3 * DBL_EPSILON)`
- `#define DLL_EXPORT`
- `#define pfi_htof32(x) (x)`
- `#define pfi_ftoh32(x) (x)`
- `#define pfi_htof16(x) (x)`
- `#define pfi_ftoh16(x) (x)`

Enumerations

- `enum { pfi_grey = 0, pfi_rgb, pfi_yuv, pfi_unknown }`

Functions

- `int pfi_getv (FILE *stream, double *valp)`
- `int pfi_putv (FILE *stream, double val)`
- `int pfi_properties (FILE *stream, int *rows, int *cols, int *colors, int *colormodel)`
- `double * pfi_read (FILE *stream, int *rows, int *cols, int *colors, int *colormodel)`
- `int pfi_write (FILE *stream, double *buf, int rows, int cols, int colors, int colormodel, char *fname)`
- `DLL_EXPORT int pfi_read_name (char *name, int rows, int cols, int colors, double *result)`
- `DLL_EXPORT int pfi_write_name (char *name, double *buf, int rows, int cols, int colors, int colormodel)`
- `DLL_EXPORT int pfi_properties_name (char *name, int *rows, int *cols, int *colors, int *colormodel)`

2.3.1 Define Documentation

2.3.1.1 #define **DLL_EXPORT**

Definition at line 36 of file pfi.h.

2.3.1.2 #define **PFI_EPSILON** (3 * DBL_EPSILON)

Definition at line 28 of file pfi.h.

2.3.1.3 #define **pfi_ftoh16**(x) (x)

Definition at line 131 of file pfi.h.

2.3.1.4 #define **pfi_ftoh32**(x) (x)

Definition at line 129 of file pfi.h.

2.3.1.5 #define **pfi_htof16**(x) (x)

Definition at line 130 of file pfi.h.

2.3.1.6 #define **pfi_htof32**(x) (x)

Definition at line 128 of file pfi.h.

2.3.1.7 #define **PFI_VERSION** "1.2"

Definition at line 26 of file pfi.h.

2.3.2 Enumeration Type Documentation

2.3.2.1 anonymous enum

Colormodel specifier. Either greyscale, rgb or yuv.

Enumerator:

pfi_grey

pfi_rgb

pfi_yuv

pfi_unknown

Definition at line 136 of file pfi.h.

2.3.3 Function Documentation

2.3.3.1 `int pfi_getv (FILE * stream, double * valp)`

Reads a double value from an open pfi file

Parameters

<i>stream</i>	The open file.
<i>valp</i>	A reference to the value to be read.

Returns

0 on success, else -1.

Definition at line 380 of file pfi.c.

References DPRINTF, pfi2double(), pfi_ftoh16, pfi_ftoh32, and PIXEL_PFIS.

2.3.3.2 `int pfi_properties (FILE * stream, int * rows, int * cols, int * colors, int * colormodel)`

Gets the pfi image properties from an open file.

Parameters

<i>stream</i>	The open file.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	A reference to an integer to represent the colormodel used in this image to be determined from the image header. We use 0 for none (greyscale), 1 for RGB etc.

Returns

success: 0, else -1.

Definition at line 489 of file pfi.c.

References DPRINTF, numexpect(), and skipcomments().

2.3.3.3 `DLL_EXPORT int pfi_properties_name (char * name, int * rows, int * cols, int * colors, int * colormodel)`

Gets the pfi image properties from an image file.

Parameters

<i>name</i>	The file name.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	A reference to an integer to represent the colormodel used in this image to be determined from the image header. We use 0 for none (greyscale), 1 for RGB etc.

Returns

success: 0, else -1.

Definition at line 888 of file pfi.c.

References pfi_properties().

2.3.3.4 int **pfi_putv** (FILE * stream, double val)

Writes a double value into an open pfi file.

Parameters

<i>stream</i>	The open file.
<i>val</i>	The value to be written.

Returns

0 on success, else -1.

Definition at line 428 of file pfi.c.

References double2pfi(), DPRINTF, pfi_htof16, pfi_htof32, and PIXEL_PFIS.

2.3.3.5 double* **pfi_read** (FILE * stream, int * rows, int * cols, int * colors, int * colormodel)

Reads a pfi image from an open file.

Parameters

<i>stream</i>	The open file.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.

<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	A reference to an integer to represent the colormodel used in this image to be determined from the image header. We use 0 for none (greyscale), 1 for RGB etc.

Returns

A new double array containing the image. The colors are stored one after another, so we've got pixel one with all colors first, then pixel two etc. In case of error, NULL is returned

Definition at line 654 of file pfi.c.

References DPRINTF, pfi_properties(), and pfi_read_internal().

2.3.3.6 **DLL_EXPORT** int **pfi_read_name** (char * name, int rows, int cols, int colors, double * result)

Reads a pfi image.

Parameters

<i>name</i>	The file name.
<i>rows</i>	the number of rows
<i>cols</i>	the number of cols
<i>colors</i>	the number of colors
<i>result</i>	A double array containing the image. The colors are stored one after another, so we've got pixel one with all colors first, then pixel two etc.

Returns

0 if successful, else -1

Definition at line 770 of file pfi.c.

References pfi_read_internal().

2.3.3.7 int **pfi_write** (FILE * stream, double * buf, int rows, int cols, int colors, int colormodel, char * fname)

Writes a pfi image to an open file.

Parameters

<i>stream</i>	The open file.
<i>buf</i>	A reference to the buffer containing the image. The colors are stored one after another, so we've got pixel one with all colors first, then pixel two etc.

<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	An integer to represent the colormodel used in this image. We use 0 for none (greyscale), 1 for RGB etc.
<i>fname</i>	The name which will be written to the file header.

Returns

0 on success, else -1.

Definition at line 715 of file pfi.c.

References `pfi_putv()`.

2.3.3.8 **DLL_EXPORT** int **pfi_write_name** (char * name, double * buf, int rows, int cols, int colors, int colormodel)

Opens and writes a pfi image.

Parameters

<i>name</i>	The file name.
<i>buf</i>	A reference to the buffer containing the image. The colors are stored one after another, so we've got pixel one with all colors first, then pixel two etc.
<i>rows</i>	A reference to the number of rows to be determined from the image header.
<i>cols</i>	A reference to the number of cols to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colors</i>	A reference to the number of colors to be determined from the image header.
<i>colormodel</i>	An integer to represent the colormodel used in this image. We use 0 for none (greyscale), 1 for RGB etc.

Returns

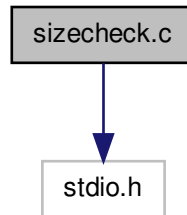
0 on success, else -1.

Definition at line 833 of file pfi.c.

References `pfi_write()`.

2.4 sizecheck.c File Reference

`#include <stdio.h>` Include dependency graph for sizecheck.c:



Functions

- `int main (void)`

2.4.1 Function Documentation

2.4.1.1 `int main (void)`

Definition at line 4 of file `sizecheck.c`.