Photo Finish

1

Generated by Doxygen 1.8.5

Sun Nov 17 2013 16:31:29

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 CMS Namespace Reference

Classes

- class Profile

  Wrap LCMS2's cmsHPROFILE.
- class Format

  Wrap LCMS2's pixel format.
- class Transform

  Wrap LCMS2's transform object.

Enumerations

- enum ColourModel {
  ColourModel::Any = 0, ColourModel::Greyscale = 3, ColourModel::RGB, ColourModel::CMY,
  ColourModel::CMYK, ColourModel::YCbCr, ColourModel::YUV, ColourModel::XYZ,
  ColourModel::Lab, ColourModel::YUVK, ColourModel::HSV, ColourModel::HLS,
  ColourModel::Yxy, ColourModel::MCH1, ColourModel::MCH2, ColourModel::MCH3,
  ColourModel::MCH4, ColourModel::MCH5, ColourModel::MCH6, ColourModel::MCH7,
  ColourModel::MCH8, ColourModel::MCH9, ColourModel::MCH10, ColourModel::MCH11,
  ColourModel::MCH12, ColourModel::MCH13, ColourModel::MCH14, ColourModel::MCH15,
  ColourModel::LabV2 }

  An enum class of LCMS2's colour models.
- enum Intent {
  Intent::Perceptual, Intent::Relative_colormetric, Intent::Saturation, Intent::Absolute_colormetric,
  Intent::Preserve_k_only_perceptual = 10, Intent::Preserve_k_only_relative_colormetric, Intent::Preserve_k_only_saturation, Intent::Preserve_k_only_absolute_colormetric,
  Intent::Preserve_k_plane_perceptual, Intent::Preserve_k_plane_relative_colormetric, Intent::Preserve_k_plane_saturation, Intent::Preserve_k_plane_absolute_colormetric }

  Wrap LCMS2's intents.

Functions

- std::ostream & operator<< (std::ostream &out, ColourModel model)
- std::ostream & operator<< (std::ostream &out, Format f)
- cmsIOHANDLER ∗ OpenIOhandlerFromIStream (std::istream ∗is)
- cmsIOHANDLER ∗ OpenIOhandlerFromIFStream (fs::path filepath)

- cmsUInt32Number istream_read (cmsIOHANDLER ∗iohandler, void ∗Buffer, cmsUInt32Number size, cmsUInt32Number count)
- cmsBool istream_seek (cmsIOHANDLER ∗iohandler, cmsUInt32Number offset)
- cmsBool istream_close (cmsIOHANDLER ∗iohandler)
- cmsUInt32Number istream_tell (cmsIOHANDLER ∗iohandler)
- cmsBool istream_write (cmsIOHANDLER ∗iohandler, cmsUInt32Number size, const void ∗Buffer)
- cmsUInt32Number ostream_read (cmsIOHANDLER ∗iohandler, void ∗Buffer, cmsUInt32Number size, cmsUInt32Number count)
- cmsBool ostream_seek (cmsIOHANDLER ∗iohandler, cmsUInt32Number offset)
- cmsBool ostream_close (cmsIOHANDLER ∗iohandler)
- cmsUInt32Number ostream_tell (cmsIOHANDLER ∗iohandler)
- cmsBool ostream_write (cmsIOHANDLER ∗iohandler, cmsUInt32Number size, const void ∗Buffer)

5.1.1 Enumeration Type Documentation

5.1.1.1 enum CMS::ColourModel [strong]

An enum class of LCMS2's colour models.

Enumerator

Any

Greyscale

RGB

CMY

CMYK

YCbCr

YUV

XYZ

Lab

YUVK

HSV

HLS

Yxy

MCH1

MCH2

MCH3

MCH4

MCH5

MCH6

MCH7

MCH8

MCH9

MCH10

MCH11

MCH12

MCH13

MCH14

MCH15

LabV2

Definition at line 93 of file CMS.hh.

5.1.1.2 enum CMS::Intent [strong]

Wrap LCMS2's intents.

Enumerator

- ***Perceptual***
- ***Relative_colormetric***
- ***Saturation***
- ***Absolute_colormetric***
- ***Preserve_k_only_perceptual***
- ***Preserve_k_only_relative_colormetric***
- ***Preserve_k_only_saturation***
- ***Preserve_k_only_absolute_colormetric***
- ***Preserve_k_plane_perceptual***
- ***Preserve_k_plane_relative_colormetric***
- ***Preserve_k_plane_saturation***
- ***Preserve_k_plane_absolute_colormetric***

Definition at line 334 of file CMS.hh.

5.1.2 Function Documentation

5.1.2.1 cmsBool CMS::istream_close (cmsIOHANDLER ∗ *iohandler*)

Definition at line 577 of file CMS.cc.

5.1.2.2 cmsUInt32Number CMS::istream_read (cmsIOHANDLER ∗ *iohandler,* void ∗ *Buffer,* cmsUInt32Number *size,* cmsUInt32Number *count*)

Definition at line 563 of file CMS.cc.

5.1.2.3 cmsBool CMS::istream_seek (cmsIOHANDLER ∗ *iohandler,* cmsUInt32Number *offset*)

Definition at line 570 of file CMS.cc.

5.1.2.4 cmsUInt32Number CMS::istream_tell (cmsIOHANDLER ∗ *iohandler*)

Definition at line 589 of file CMS.cc.

5.1.2.5 cmsBool CMS::istream_write (cmsIOHANDLER ∗ *iohandler,* cmsUInt32Number *size,* const void ∗ *Buffer*)

Definition at line 594 of file CMS.cc.

5.1.2.6 cmsIOHANDLER ∗ CMS::OpenIOhandlerFromIFStream (fs::path *filepath*)

Definition at line 554 of file CMS.cc.

5.1.2.7 cmsIOHANDLER ∗ CMS::OpenIOhandlerFromIStream (std::istream ∗ *is*)

Definition at line 532 of file CMS.cc.

5.1.2.8 std::ostream & CMS::operator<< (std::ostream & *out,* ColourModel *model*)

Definition at line 159 of file CMS.cc.

5.1.2.9 std::ostream & CMS::operator<< (std::ostream & *out,* Format *f*)

Definition at line 428 of file CMS.cc.

5.1.2.10 cmsBool CMS::ostream_close (cmsIOHANDLER ∗ *iohandler*)

Definition at line 611 of file CMS.cc.

5.1.2.11 cmsUInt32Number CMS::ostream_read (cmsIOHANDLER ∗ *iohandler,* void ∗ *Buffer,* cmsUInt32Number *size,* cmsUInt32Number *count*)

Definition at line 599 of file CMS.cc.

5.1.2.12 cmsBool CMS::ostream_seek (cmsIOHANDLER ∗ *iohandler,* cmsUInt32Number *offset*)

Definition at line 604 of file CMS.cc.

5.1.2.13 cmsUInt32Number CMS::ostream_tell (cmsIOHANDLER ∗ *iohandler*)

Definition at line 623 of file CMS.cc.

5.1.2.14 cmsBool CMS::ostream_write (cmsIOHANDLER ∗ *iohandler,* cmsUInt32Number *size,* const void ∗ *Buffer*)

Definition at line 628 of file CMS.cc.

5.2 PhotoFinish Namespace Reference

Classes

- class CropSolver

  Class for finding the best frame position for cropping.
- class definable

  Template class for storing things that can be defined or undefined.
- class Role_Definable

  Base class for adding "definable" attribute.
- class Destination

  Represents a destination, read from destinations.yml.
- class Destinations

  A wrapper class for reading destinations from a YAML file and storing them in a map.
- class D_sharpen

  Sharpen parameters for destination.
- class D_resize

  Resize parameters for destination.
- class D_target

  Target parameters for destination.

- class D_JPEG

  JPEG parameters for destination.
- class D_PNG

  PNG parameters for destination.
- class D_TIFF

  TIFF parameters for destination.
- class D_JP2

  JP2 parameters for destination.
- class D_WebP

  WebP parameters for destination.
- class D_profile

  ICC profile parameters for destination.
- class D_thumbnail

  Thumbnail parameters for destination.
- class Ditherer

  Class for dithering images down to 8-bit components.
- class Uninitialised

  Uninitialised attribute exception.
- class Unimplemented

  Unimplemented method exception.
- class NoResults

  No results exception.
- class NoTargets

  No targets exception.
- class ErrorMsg

  Generic error message exception.
- class MemAllocError

  Memory allocation exception.
- class FileError

  File error abstract base exception.
- class UnknownFileType

  Unknown file type exception.
- class FileOpenError

  File open exception.
- class FileContentError

  File content exception.
- class DestinationError

  Destination exception.
- class LibraryError

  Library exception.
- class cmsTypeError
- class WebPError

  WebP exception.
- class Frame

  Crop+rescaling parameters.
- class Image

  An image class.
- class ImageFilepath

  Class for holding filename and the image format.
- class ImageReader

  Abstract base class for reading image files.

- class ImageWriter

  Abstract base class for writing image files.
- class SOLwriter

  Write the boot logo files for use on Motorola Atrix 4G and possibly other phones.
- class Kernel1Dvar

  Creates and stores coefficients for cropping and resizing an image.
- class Lanczos

  Lanczos filter.
- class Kernel2D

  Creates and stores coefficients for convolving an image.
- class GaussianSharpen

  GaussianSharpen kernel.
- class Kernel2Dvar

  Creates and stores coefficients for cropping and resizing an image.
- class Lanczos2D

  Lanczos filter.
- class Tags

  Reads and holds tag information.
- struct jpeg_source_state_t

  Structure holding information for the istream reader.
- struct jpeg_destination_state_t

  Structure holding information for the ostream writer.
- struct PNGreader_cb
- class StreamIO
- class webp_stream_writer

  A custom writer for libwebp that writes using a std::ostream object.

Typedefs

- typedef std::pair< double, double > rulerpair

  Ruler paramaters - percentage of final image vs. pixel position in original.
- typedef std::list< rulerpair > rulerlist

  A list of rulers.
- typedef std::map< std::string,
  std::string > hash

  A simple hash.
- typedef std::vector< std::string > stringlist

  A list of strings.
- typedef std::map< std::string,
  stringlist > multihash

  A hash of string lists.
- typedef std::vector< std::pair
  < std::string, std::string > > subst_table

Functions

- template<typename T >
  T scaleval (void)

  A template function that returns the 'scale' value of a type.
- template<>
  unsigned char scaleval< unsigned char > (void)

- template<>
  unsigned short int scaleval< unsigned short int > (void)
- template<>
  unsigned int scaleval< unsigned int > (void)
- template<>
  unsigned long int scaleval< unsigned long int > (void)
- template<>
  float scaleval< float > (void)
- template<>
  double scaleval< double > (void)
- template<typename T >
  T limitval (SAMPLE v)

  A template function that limits a floating-point value while converting to another type.
- template<>
  unsigned char limitval< unsigned char > (SAMPLE v)
- template<>
  unsigned short int limitval< unsigned short int > (SAMPLE v)
- template<>
  unsigned int limitval< unsigned int > (SAMPLE v)
- template<>
  unsigned long int limitval< unsigned long int > (SAMPLE v)
- template<>
  float limitval< float > (SAMPLE v)
- template<>
  double limitval< double > (SAMPLE v)
- bool exists (const ImageFilepath &fp)
- std::time_t last_write_time (const ImageFilepath &fp)
- template<typename Num_type , typename R_type >
  Exiv2::ValueType< R_type > & closest_Rational (double value)

  Find a close rational fraction given a floating-point value.
- short unsigned int read_be16 (const unsigned char ∗data)
- unsigned int read_be32 (const unsigned char ∗data)
- void jpeg_read_metadata (jpeg_decompress_struct ∗dinfo, Image::ptr img)
- void add_rulers (multihash &vars, std::string key, rulerlist &rulers)

  Parse named variables into a list of rulers.
- void add_ruler_pins (rulerlist &rulers, unsigned int max)

  Add rulers to the either side of an image if there aren't enough.
- void error_callback (const char ∗msg, void ∗client_data)

  Error callback for OpenJPEG - throw a LibraryError exception.
- void warning_callback (const char ∗msg, void ∗client_data)

  Warning callback for OpenJPEG - print the message to STDERR.
- void info_callback (const char ∗msg, void ∗client_data)

  Info callback for OpenJPEG - print the indented message to STDERR.
- template<typename T >
  void read_planar (unsigned int width, unsigned char channels, opj_image_t ∗image, T ∗row, unsigned int y)

  Read a row of image data from OpenJPEG's planar integer components into an LCMS2-compatible single array.
- template<typename T >
  void write_planar (unsigned int width, unsigned char channels, T ∗row, opj_image_t ∗image, unsigned int y)

  Read a row of planar pixel data into OpenJPEG's planar components.
- template<typename T >
  void write_packed (unsigned int width, unsigned char channels, T ∗row, opj_image_t ∗image, unsigned int y)

  Read a row of packed pixel data into OpenJPEG's planar components.
- void jpeg_istream_src (j_decompress_ptr dinfo, std::istream ∗is)

  Set up a "source manager" on the given JPEG decompression structure to read from an istream.

- void jpeg_istream_src_free (j_decompress_ptr dinfo)

  Free the data structures of the istream source manager.
- void jpeg_ostream_dest (j_compress_ptr cinfo, std::ostream ∗os)

  Setup a "destination manager" on the given JPEG compression structure to write to an ostream.
- void jpeg_ostream_dest_free (j_compress_ptr cinfo)

  Free the data structures of the ostream destination manager.
- void jpegfile_scan_RGB (jpeg_compress_struct ∗cinfo)

  Create a scan "script" for an RGB image.
- void jpegfile_scan_greyscale (jpeg_compress_struct ∗cinfo)

  Create a scan "script" for a greyscale image.
- CMS::Profile::ptr jpeg_read_profile (jpeg_decompress_struct ∗dinfo, Destination::ptr dest)

  Read an ICC profile from APP2 markers in a JPEG file.
- void jpeg_write_profile (jpeg_compress_struct ∗cinfo, unsigned char ∗data, unsigned int size)

  Write an ICC profile into APP2 markers in a JPEG file.
- void jpeg_istream_init_source (j_decompress_ptr dinfo)

  Initialise the istream source manager.
- boolean jpeg_istream_fill_input_buffer (j_decompress_ptr dinfo)

  Fill the buffer.
- void jpeg_istream_skip_input_data (j_decompress_ptr dinfo, long num_bytes)

  Skip some data.
- boolean jpeg_istream_resync_to_restart (j_decompress_ptr dinfo, int desired)

  Resync to start?!?
- void jpeg_istream_term_source (j_decompress_ptr dinfo)

  Terminate the istream source manager.
- void png_read_metadata (png_structp png, png_infop info, Image::ptr image)
- void png_info_cb (png_structp png, png_infop info)

  Called by libPNG when the iHDR chunk has been read with the main "header" information.
- void png_row_cb (png_structp png, png_bytep row_data, png_uint_32 row_num, int pass)

  Called by libPNG when a row of image data has been read.
- void png_end_cb (png_structp png, png_infop info)

  Called by libPNG when the image data has finished.
- void png_write_ostream_cb (png_structp png, png_bytep buffer, png_size_t length)

  libPNG callback for writing to an ostream
- void png_flush_ostream_cb (png_structp png)

  libPNG callback for flushing an ostream
- void write_be (void ∗ptr, size_t size, std::ostream &stream)
- int webp_stream_writer_func (const uint8_t ∗data, size_t data_size, const WebPPicture ∗picture)

  Wrapper around the webp_stream_writer class.
- void copy_le_to (unsigned char ∗dest, unsigned int value, unsigned char length)
- unsigned int read_le32 (const unsigned char ∗data)
- template<typename A , typename B >
  void transfer_alpha_typed2 (unsigned int width, unsigned char src_channels, const A ∗src_row, unsigned char dest_channels, const B ∗dest_row)
- template<typename A >
  void transfer_alpha_typed (unsigned int width, unsigned char src_channels, const A ∗src_row, CMS::Format dest_format, const void ∗dest_row)
- void transfer_alpha (unsigned int width, CMS::Format src_format, const void ∗src_row, CMS::Format dest_format, const void ∗dest_row)
- std::string profile_name (CMS::Profile::ptr profile)
- void lcms2_errorhandler (cmsContext ContextID, cmsUInt32Number ErrorCode, const char ∗Text)

  Throw a LibraryError exception whem LCMS2 returns an error.
- void lcms2_error_adaptor (void)

Set up an error handler with LCMS2 that will throw a LibraryError exception.

- Exiv2::ExifKey exif_key_read (std::string key_string)
- Exiv2::Value::AutoPtr exif_value_read (Exiv2::ExifKey key, std::string value_string)

  Read an EXIF value from a string, with optional substitution for enum-style values.
- Exiv2::IptcKey iptc_key_read (std::string key_string)
- Exiv2::XmpKey xmp_key_read (std::string key_string)
- template<typename Num_type , typename R_type >
  Exiv2::Value::AutoPtr parse_Rational (std::string s)

  Parse a string into a rational fraction.

Variables

- unsigned char header [12]
- std::map< std::string, WebPPreset > WebP_presets
- subst_table EXIF_key_subst

  Map from Image::Exiftool tag names to Exiv2's tag names.
- std::map< std::string,
  subst_table > EXIF_value_subst
- subst_table IPTC_key_subst

  Map from Image::Exiftool tag names to Exiv2's tag names.
- subst_table XMP_key_subst

  Map from Image::Exiftool tag names to Exiv2's tag names.

5.2.1 Typedef Documentation

5.2.1.1 typedef std::map<std::string, std::string> PhotoFinish::hash

A simple hash.

Definition at line 36 of file Destination_items.hh.

5.2.1.2 typedef std::map<std::string, stringlist > PhotoFinish::multihash

A hash of string lists.

Definition at line 42 of file Destination_items.hh.

5.2.1.3 typedef std::list< rulerpair > PhotoFinish::rulerlist

A list of rulers.

Definition at line 36 of file CropSolution.hh.

5.2.1.4 typedef std::pair<double, double> PhotoFinish::rulerpair

Ruler paramaters - percentage of final image vs. pixel position in original.

Definition at line 33 of file CropSolution.hh.

5.2.1.5 typedef std::vector<std::string> PhotoFinish::stringlist

A list of strings.

Definition at line 39 of file Destination_items.hh.

5.2.1.6 typedef std::vector<std::pair<std::string, std::string> > PhotoFinish::subst_table

Definition at line 37 of file Tags.hh.

5.2.2 Function Documentation

5.2.2.1 void PhotoFinish::add_ruler_pins (rulerlist & *rulers,* unsigned int *max*)

Add rulers to the either side of an image if there aren't enough.

Definition at line 53 of file CropSolution.cc.

5.2.2.2 void PhotoFinish::add_rulers (multihash & *vars,* std::string *key,* rulerlist & *rulers*)

Parse named variables into a list of rulers.

Parameters

vars	The tag variables
key	The name of the variables to parse
rulers	The list of rulers to add to

Definition at line 36 of file CropSolution.cc.

5.2.2.3 template<typename Num_type , typename R_type > Exiv2::ValueType<R_type>& PhotoFinish::closest_Rational (double *value*)

Find a close rational fraction given a floating-point value.

Definition at line 101 of file Tags.hh.

5.2.2.4 void PhotoFinish::copy_le_to (unsigned char ∗ *dest,* unsigned int *value,* unsigned char *length*) `[inline]`

Definition at line 95 of file WebP_ostream.hh.

5.2.2.5 void PhotoFinish::error_callback (const char ∗ *msg,* void ∗ *client_data*)

Error callback for OpenJPEG - throw a LibraryError exception.

Definition at line 25 of file JP2_callbacks.cc.

5.2.2.6 Exiv2::ExifKey PhotoFinish::exif_key_read (std::string *key_string*)

Definition at line 81 of file Tags_EXIF_subst.cc.

5.2.2.7 Exiv2::Value::AutoPtr PhotoFinish::exif_value_read (Exiv2::ExifKey *key,* std::string *value_string*)

Read an EXIF value from a string, with optional substitution for enum-style values.

Definition at line 280 of file Tags_EXIF_subst.cc.

5.2.2.8 bool PhotoFinish::exists (const ImageFilepath & *fp*) `[inline]`

Definition at line 92 of file ImageFile.hh.

5.2.2.9 void PhotoFinish::info_callback (const char ∗ *msg,* void ∗ *client_data*)

Info callback for OpenJPEG - print the indented message to STDERR.

Definition at line 34 of file JP2_callbacks.cc.

5.2.2.10 Exiv2::IptcKey PhotoFinish::iptc_key_read (std::string *key_string*)

Definition at line 37 of file Tags_IPTC_subst.cc.

5.2.2.11 boolean PhotoFinish::jpeg_istream_fill_input_buffer (j_decompress_ptr *dinfo*)

Fill the buffer.

Definition at line 45 of file JPEG_iostream.cc.

5.2.2.12 void PhotoFinish::jpeg_istream_init_source (j_decompress_ptr *dinfo*)

Initialise the istream source manager.

Definition at line 34 of file JPEG_iostream.cc.

5.2.2.13 boolean PhotoFinish::jpeg_istream_resync_to_restart (j_decompress_ptr *dinfo,* int *desired*)

Resync to start?!?

Definition at line 74 of file JPEG_iostream.cc.

5.2.2.14 void PhotoFinish::jpeg_istream_skip_input_data (j_decompress_ptr *dinfo,* long *num_bytes*)

Skip some data.

Definition at line 57 of file JPEG_iostream.cc.

5.2.2.15 void PhotoFinish::jpeg_istream_src (j_decompress_ptr *dinfo,* std::istream ∗ *is*)

Set up a "source manager" on the given JPEG decompression structure to read from an istream.

Definition at line 84 of file JPEG_iostream.cc.

5.2.2.16 void PhotoFinish::jpeg_istream_src_free (j_decompress_ptr *dinfo*)

Free the data structures of the istream source manager.

Definition at line 99 of file JPEG_iostream.cc.

5.2.2.17 void PhotoFinish::jpeg_istream_term_source (j_decompress_ptr *dinfo*)

Terminate the istream source manager.

Definition at line 79 of file JPEG_iostream.cc.

5.2.2.18 void PhotoFinish::jpeg_ostream_dest (j_compress_ptr *cinfo,* std::ostream ∗ *os*)

Setup a "destination manager" on the given JPEG compression structure to write to an ostream.

Definition at line 144 of file JPEG_iostream.cc.

5.2.2.19 void PhotoFinish::jpeg_ostream_dest_free (j_compress_ptr *cinfo*)

Free the data structures of the ostream destination manager.

Definition at line 157 of file JPEG_iostream.cc.

5.2.2.20 void PhotoFinish::jpeg_read_metadata (jpeg_decompress_struct ∗ *dinfo,* Image::ptr *img*)

Definition at line 32 of file JPEG_metadata.cc.

5.2.2.21 CMS::Profile::ptr PhotoFinish::jpeg_read_profile (jpeg_decompress_struct ∗ *dinfo,* Destination::ptr *dest*)

Read an ICC profile from APP2 markers in a JPEG file.

Definition at line 31 of file JPEG_profiles.cc.

5.2.2.22 void PhotoFinish::jpeg_write_profile (jpeg_compress_struct ∗ *cinfo,* unsigned char ∗ *data,* unsigned int *size*)

Write an ICC profile into APP2 markers in a JPEG file.

Definition at line 78 of file JPEG_profiles.cc.

5.2.2.23 void PhotoFinish::jpegfile_scan_greyscale (jpeg_compress_struct ∗ *cinfo*)

Create a scan "script" for a greyscale image.

Create a scan script for encoding a greyscale progressive JPEG.

Definition at line 114 of file JPEG_scans.cc.

5.2.2.24 void PhotoFinish::jpegfile_scan_RGB (jpeg_compress_struct ∗ *cinfo*)

Create a scan "script" for an RGB image.

Create a scan script for encoding a colour progressive JPEG.

Definition at line 26 of file JPEG_scans.cc.

5.2.2.25 std::time_t PhotoFinish::last_write_time (const ImageFilepath & *fp*) `[inline]`

Definition at line 93 of file ImageFile.hh.

5.2.2.26 void PhotoFinish::lcms2_error_adaptor (void)

Set up an error handler with LCMS2 that will throw a LibraryError exception.

Definition at line 29 of file LCMS2ErrorHandler.cc.

5.2.2.27 void PhotoFinish::lcms2_errorhandler (cmsContext *ContextID*, cmsUInt32Number *ErrorCode*, const char ∗ *Text*)

Throw a LibraryError exception whem LCMS2 returns an error.

Definition at line 25 of file LCMS2ErrorHandler.cc.

5.2.2.28 template<typename T > T PhotoFinish::limitval (SAMPLE *v*)

A template function that limits a floating-point value while converting to another type.

5.2.2.29 template<> double PhotoFinish::limitval< double > (SAMPLE *v*) [inline]

Definition at line 250 of file Image.hh.

5.2.2.30 template<> float PhotoFinish::limitval< float > (SAMPLE *v*) [inline]

Definition at line 245 of file Image.hh.

5.2.2.31 template<> unsigned char PhotoFinish::limitval< unsigned char > (SAMPLE *v*) [inline]

Definition at line 209 of file Image.hh.

5.2.2.32 template<> unsigned int PhotoFinish::limitval< unsigned int > (SAMPLE *v*) [inline]

Definition at line 227 of file Image.hh.

5.2.2.33 template<> unsigned long int PhotoFinish::limitval< unsigned long int > (SAMPLE *v*) [inline]

Definition at line 236 of file Image.hh.

5.2.2.34 template<> unsigned short int PhotoFinish::limitval< unsigned short int > (SAMPLE *v*) [inline]

Definition at line 218 of file Image.hh.

5.2.2.35 template<typename Num_type , typename R_type > Exiv2::Value::AutoPtr PhotoFinish::parse_Rational (std::string *s*)

Parse a string into a rational fraction.

Definition at line 267 of file Tags_EXIF_subst.cc.

5.2.2.36 void PhotoFinish::png_end_cb (png_structp *png*, png_infop *info*)

Called by libPNG when the image data has finished.

Definition at line 135 of file PNGreader_cb.cc.

5.2.2.37 void PhotoFinish::png_flush_ostream_cb (png_structp *png*)

libPNG callback for flushing an ostream

Definition at line 65 of file PNGwriter.cc.

5.2.2.38 void PhotoFinish::png_info_cb (png_structp *png*, png_infop *info*)

Called by libPNG when the iHDR chunk has been read with the main "header" information.

Definition at line 115 of file PNGreader_cb.cc.

5.2.2.39 void PhotoFinish::png_read_metadata (png_structp *png*, png_infop *info*, Image::ptr *image*)

Definition at line 24 of file PNG_metadata.cc.

5.2.2.40 void PhotoFinish::png_row_cb (png_structp *png*, png_bytep *row_data*, png_uint_32 *row_num*, int *pass*)

Called by libPNG when a row of image data has been read.

Definition at line 126 of file PNGreader_cb.cc.

5.2.2.41 void PhotoFinish::png_write_ostream_cb (png_structp *png*, png_bytep *buffer*, png_size_t *length*)

libPNG callback for writing to an ostream

Definition at line 59 of file PNGwriter.cc.

5.2.2.42 std::string PhotoFinish::profile_name (CMS::Profile::ptr *profile*)

Definition at line 142 of file Image.cc.

5.2.2.43 short unsigned int PhotoFinish::read_be16 (const unsigned char ∗ *data*) [inline]

Definition at line 24 of file JPEG_metadata.cc.

5.2.2.44 unsigned int PhotoFinish::read_be32 (const unsigned char ∗ *data*) [inline]

Definition at line 28 of file JPEG_metadata.cc.

5.2.2.45 unsigned int PhotoFinish::read_le32 (const unsigned char ∗ *data*) [inline]

Definition at line 102 of file WebP_ostream.hh.

5.2.2.46 template<typename T > void PhotoFinish::read_planar (unsigned int *width*, unsigned char *channels*, opj_image_t ∗ *image*, T ∗ *row*, unsigned int *y*) [inline]

Read a row of image data from OpenJPEG's planar integer components into an LCMS2-compatible single array.

Definition at line 36 of file JP2.hh.

5.2.2.47 template<typename T > T PhotoFinish::scaleval (void)

A template function that returns the 'scale' value of a type.

5.2.2.48 template<> double PhotoFinish::scaleval< double > (void) [inline]

Definition at line 201 of file Image.hh.

5.2.2.49 template<> float **PhotoFinish::scaleval**< float >(void) [inline]

Definition at line 198 of file Image.hh.

5.2.2.50 template<> unsigned char **PhotoFinish::scaleval**< unsigned char >(void) [inline]

Definition at line 186 of file Image.hh.

5.2.2.51 template<> unsigned int **PhotoFinish::scaleval**< unsigned int >(void) [inline]

Definition at line 192 of file Image.hh.

5.2.2.52 template<> unsigned long int **PhotoFinish::scaleval**< unsigned long int >(void) [inline]

Definition at line 195 of file Image.hh.

5.2.2.53 template<> unsigned short int **PhotoFinish::scaleval**< unsigned short int >(void) [inline]

Definition at line 189 of file Image.hh.

5.2.2.54 void PhotoFinish::transfer_alpha (unsigned int *width*, **CMS::Format** *src_format*, const void * *src_row*, **CMS::Format** *dest_format*, const void * *dest_row*)

Definition at line 117 of file Image.cc.

5.2.2.55 template<typename A > void PhotoFinish::transfer_alpha_typed (unsigned int *width*, unsigned char *src_channels*, const A * *src_row*, **CMS::Format** *dest_format*, const void * *dest_row*)

Definition at line 91 of file Image.cc.

5.2.2.56 template<typename A , typename B > void PhotoFinish::transfer_alpha_typed2 (unsigned int *width*, unsigned char *src_channels*, const A * *src_row*, unsigned char *dest_channels*, const B * *dest_row*)

Definition at line 82 of file Image.cc.

5.2.2.57 void PhotoFinish::warning_callback (const char * *msg*, void * *client_data*)

Warning callback for OpenJPEG - print the message to STDERR.

Definition at line 29 of file JP2_callbacks.cc.

5.2.2.58 int PhotoFinish::webp_stream_writer_func (const uint8_t * *data*, size_t *data_size*, const WebPPicture * *picture*)

Wrapper around the webp_stream_writer class.

Definition at line 187 of file WebP_ostream.cc.

5.2.2.59 void PhotoFinish::write_be (void * *ptr*, size_t *size*, std::ostream & *stream*)

Definition at line 46 of file SOLwriter.cc.

5.2.2.60 template<typename T > void PhotoFinish::write_packed (unsigned int *width,* unsigned char *channels,* T ∗ *row,* opj_image_t ∗ *image,* unsigned int *y*)

Read a row of packed pixel data into OpenJPEG's planar components.

Definition at line 60 of file JP2.hh.

5.2.2.61 template<typename T > void PhotoFinish::write_planar (unsigned int *width,* unsigned char *channels,* T ∗ *row,* opj_image_t ∗ *image,* unsigned int *y*)

Read a row of planar pixel data into OpenJPEG's planar components.

Definition at line 48 of file JP2.hh.

5.2.2.62 Exiv2::XmpKey PhotoFinish::xmp_key_read (std::string *key_string*)

Definition at line 43 of file Tags_XMP_subst.cc.

5.2.3 Variable Documentation

5.2.3.1 subst_table PhotoFinish::EXIF_key_subst

Map from Image::Exiftool tag names to Exiv2's tag names.

Definition at line 27 of file Tags_EXIF_subst.cc.

5.2.3.2 std::map<std::string, subst_table> PhotoFinish::EXIF_value_subst

Definition at line 93 of file Tags_EXIF_subst.cc.

5.2.3.3 unsigned char PhotoFinish::header[12]

Initial value:

```
= { 0x53, 0x4f, 0x4c, 0x3a, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00 }
```

Definition at line 43 of file SOLwriter.cc.

5.2.3.4 subst_table PhotoFinish::IPTC_key_subst

Initial value:

```
= {
    StrPair("IPTC:By-line",                         "Iptc.Application2.Byline"),
    StrPair("IPTC:City",                    "Iptc.Application2.City"),
    StrPair("IPTC:Country-PrimaryLocationCode",   "Iptc.Application2.CountryCode"),
    StrPair("IPTC:Country-PrimaryLocationName",   "Iptc.Application2.CountryName"),
    StrPair("IPTC:CopyrightNotice",               "Iptc.Application2.Copyright"),
    StrPair("IPTC:Province-State",                "Iptc.Application2.ProvinceState"),
    StrPair("IPTC:Sub-location",          "Iptc.Application2.SubLocation"),
  }
```

Map from Image::Exiftool tag names to Exiv2's tag names.

Definition at line 27 of file Tags_IPTC_subst.cc.

5.2.3.5 std::map<std::string, WebPPreset> PhotoFinish::WebP_presets

Initial value:

```
= { std::make_pair("Default", WEBP_PRESET_DEFAULT),
                                                     std::make_pair("Picture", WEBP_PRESET_PICTURE),
                                                     std::make_pair("Photo", WEBP_PRESET_PHOTO),
                                                     std::make_pair("Drawing", WEBP_PRESET_DRAWING),
                                                     std::make_pair("Icon", WEBP_PRESET_ICON),
                                                     std::make_pair("Text", WEBP_PRESET_TEXT) }
```

Definition at line 28 of file WebPwriter.cc.

5.2.3.6 subst_table PhotoFinish::XMP_key_subst

Initial value:

```
= {
    StrPair("XMP:Copyright",                          "Xmp.dc.Copyright"),
    StrPair("XMP:Creator",                            "Xmp.dc.Creator"),

    StrPair("XMP:CreatorContactInfoCiAdrCity",        "Xmp.iptc.CiAdrCity"),
    StrPair("XMP:CreatorContactInfoCiAdrCtry",        "Xmp.iptc.CiAdrCtry"),
    StrPair("XMP:CreatorContactInfoCiAdrExtadr", "Xmp.iptc.CiAdrExtadr"),
    StrPair("XMP:CreatorContactInfoCiAdrPcode",       "Xmp.iptc.CiAdrPcode"),

    StrPair("XMP-cc:License",                         "Xmp.cc.License"),

    StrPair("XMP-microsoft:CameraSerialNumber",       "Xmp.MicrosoftPhoto.CameraSerialNumber"),
    StrPair("XMP-microsoft:LensManufacturer",         "Xmp.MicrosoftPhoto.LensManufacturer"),
    StrPair("XMP-microsoft:LensModel",                "Xmp.MicrosoftPhoto.LensModel"),
  }
```

Map from Image::Exiftool tag names to Exiv2's tag names.

Definition at line 27 of file Tags_XMP_subst.cc.

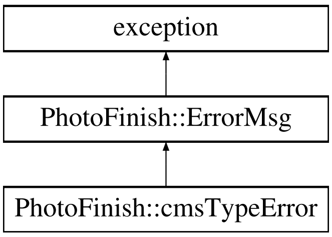
Chapter 6

Class Documentation

6.1 PhotoFinish::cmsTypeError Class Reference

`#include <Exception.hh>`

Inheritance diagram for PhotoFinish::cmsTypeError:



Public Member Functions

- cmsTypeError (const std::string &m, const unsigned int &t)

  Constructor.
- virtual const char $*$ what () const throw ()

Additional Inherited Members

6.1.1 Detailed Description

Definition at line 302 of file Exception.hh.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 PhotoFinish::cmsTypeError::cmsTypeError (const std::string & *m*, const unsigned int & *t*) `[inline]`

Constructor.

Parameters

m	Message string.
t	LCMS2 type.

Definition at line 312 of file Exception.hh.

6.1.3 Member Function Documentation

6.1.3.1 virtual const char∗ PhotoFinish::cmsTypeError::what ( ) const throw) [inline], [virtual]

Implements PhotoFinish::ErrorMsg.

Definition at line 316 of file Exception.hh.

The documentation for this class was generated from the following file:

- Exception.hh

6.2 PhotoFinish::CropSolver Class Reference

Class for finding the best frame position for cropping.

`#include <CropSolution.hh>`

Public Member Functions

- CropSolver (multihash &vars)
- Frame::ptr solve (Image::ptr img, D_target::ptr target)

6.2.1 Detailed Description

Class for finding the best frame position for cropping.

Definition at line 39 of file CropSolution.hh.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 PhotoFinish::CropSolver::CropSolver (multihash & *vars*)

Definition at line 47 of file CropSolution.cc.

6.2.3 Member Function Documentation

6.2.3.1 Frame::ptr PhotoFinish::CropSolver::solve (Image::ptr *img*, D_target::ptr *target*)

Definition at line 68 of file CropSolution.cc.

The documentation for this class was generated from the following files:

- CropSolution.hh
- CropSolution.cc

6.3 d2vector Union Reference

`#include <vector.h>`

Public Attributes

- v2df v
- double e [2]

6.3.1 Detailed Description

Definition at line 49 of file vector.h.

6.3.2 Member Data Documentation

6.3.2.1 double d2vector::e[2]

Definition at line 52 of file vector.h.

6.3.2.2 v2df d2vector::v

Definition at line 51 of file vector.h.

The documentation for this union was generated from the following file:

- vector.h

6.4 d4vector Union Reference

```
#include <vector.h>
```

Public Attributes

- v4df v
- double e [4]

6.4.1 Detailed Description

Definition at line 60 of file vector.h.

6.4.2 Member Data Documentation

6.4.2.1 double d4vector::e[4]

Definition at line 63 of file vector.h.

6.4.2.2 v4df d4vector::v

Definition at line 62 of file vector.h.

The documentation for this union was generated from the following file:

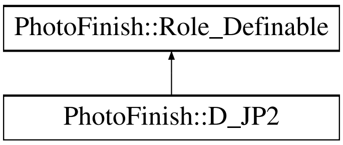
- vector.h

6.5 PhotoFinish::D_JP2 Class Reference

JP2 parameters for destination.

`#include <Destination_items.hh>`

Inheritance diagram for PhotoFinish::D_JP2:



Public Member Functions

- D_JP2 ()

  Empty constructor.
- void add_variables (multihash &vars)

  Set values from a map of "variables".
- definable< int > numresolutions (void) const
- void set_numresolutions (int n)
- definable< std::string > prog_order (void) const
- void set_prog_order (const std::string &po)
- int num_rates (void) const
- float rate (int n) const
- void set_rate (int n, float r)
- void set_rates (std::vector< float > r)
- definable< std::pair< int, int > > tile_size (void) const
- void set_tile_size (int h, int v)
- void read_config (const YAML::Node &node)

  Read a D_JP2 record from a YAML file.

Additional Inherited Members

6.5.1 Detailed Description

JP2 parameters for destination.

Definition at line 180 of file Destination_items.hh.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 PhotoFinish::D_JP2::D_JP2 ( )

Empty constructor.

Definition at line 218 of file Destination_items.cc.

6.5.3 Member Function Documentation

6.5.3.1 void PhotoFinish::D_JP2::add_variables (**multihash** & *vars*)

Set values from a map of "variables".

Definition at line 222 of file Destination_items.cc.

6.5.3.2 int PhotoFinish::D_JP2::num_rates (void) const [inline]

Definition at line 200 of file Destination_items.hh.

6.5.3.3 definable<int> PhotoFinish::D_JP2::numresolutions (void) const [inline]

Definition at line 194 of file Destination_items.hh.

6.5.3.4 definable<std::string> PhotoFinish::D_JP2::prog_order (void) const [inline]

Definition at line 197 of file Destination_items.hh.

6.5.3.5 float PhotoFinish::D_JP2::rate (int *n*) const [inline]

Definition at line 201 of file Destination_items.hh.

6.5.3.6 void PhotoFinish::D_JP2::read_config (const YAML::Node & *node*)

Read a D_JP2 record from a YAML file.

Definition at line 287 of file Destination_items.cc.

6.5.3.7 void PhotoFinish::D_JP2::set_numresolutions (int *n*) [inline]

Definition at line 195 of file Destination_items.hh.

6.5.3.8 void PhotoFinish::D_JP2::set_prog_order (const std::string & *po*) [inline]

Definition at line 198 of file Destination_items.hh.

6.5.3.9 void PhotoFinish::D_JP2::set_rate (int *n,* float *r*) [inline]

Definition at line 202 of file Destination_items.hh.

6.5.3.10 void PhotoFinish::D_JP2::set_rates (std::vector< float > *r*) [inline]

Definition at line 203 of file Destination_items.hh.

6.5.3.11 void PhotoFinish::D_JP2::set_tile_size (int *h,* int *v*) [inline]

Definition at line 206 of file Destination_items.hh.

6.5.3.12 definable< std::pair<int, int> > PhotoFinish::D_JP2::tile_size (void) const [inline]

Definition at line 205 of file Destination_items.hh.

The documentation for this class was generated from the following files:

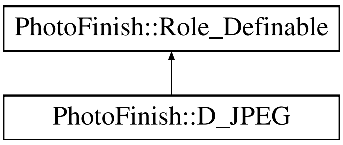
- Destination_items.hh
- Destination_items.cc

6.6 PhotoFinish::D_JPEG Class Reference

JPEG parameters for destination.

`#include <Destination_items.hh>`

Inheritance diagram for PhotoFinish::D_JPEG:



Public Member Functions

- D_JPEG ()

  Empty constructor.
- D_JPEG (int q, char h, char v, bool p)

  Constructor.
- void add_variables (multihash &vars)

  Set values from a map of "variables".
- definable< int > quality (void) const
- void set_quality (int q)
- definable< std::pair< int, int > > sample (void) const
- void set_sample (int h, int v)
- definable< bool > progressive (void) const
- void set_progressive (bool p=true)
- void read_config (const YAML::Node &node)

  Read a D_JPEG record from a YAML file.

Additional Inherited Members

6.6.1 Detailed Description

JPEG parameters for destination.

Definition at line 105 of file Destination_items.hh.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 PhotoFinish::D_JPEG::D_JPEG ( )

Empty constructor.

Definition at line 91 of file Destination_items.cc.

6.6.2.2 PhotoFinish::D_JPEG::D_JPEG (int *q*, char *h*, char *v*, bool *p*)

Constructor.

Parameters

q	Quality
h,v	Chroma sampling
p	Progressive

Definition at line 94 of file Destination_items.cc.

6.6.3 Member Function Documentation

6.6.3.1 void PhotoFinish::D_JPEG::add_variables (multihash & *vars*)

Set values from a map of "variables".

Definition at line 100 of file Destination_items.cc.

6.6.3.2 definable<bool> PhotoFinish::D_JPEG::progressive (void) const [inline]

Definition at line 132 of file Destination_items.hh.

6.6.3.3 definable<int> PhotoFinish::D_JPEG::quality (void) const [inline]

Definition at line 126 of file Destination_items.hh.

6.6.3.4 void PhotoFinish::D_JPEG::read_config (const YAML::Node & *node*)

Read a D_JPEG record from a YAML file.

Definition at line 131 of file Destination_items.cc.

6.6.3.5 definable< std::pair<int, int> > PhotoFinish::D_JPEG::sample (void) const [inline]

Definition at line 129 of file Destination_items.hh.

6.6.3.6 void PhotoFinish::D_JPEG::set_progressive (bool *p* = true) [inline]

Definition at line 133 of file Destination_items.hh.

6.6.3.7 void PhotoFinish::D_JPEG::set_quality (int *q*) [inline]

Definition at line 127 of file Destination_items.hh.

6.6.3.8 void PhotoFinish::D_JPEG::set_sample (int *h*, int *v*) [inline]

Definition at line 130 of file Destination_items.hh.

The documentation for this class was generated from the following files:

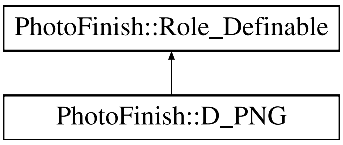
- Destination_items.hh
- Destination_items.cc

6.7 PhotoFinish::D_PNG Class Reference

PNG parameters for destination.

```
#include <Destination_items.hh>
```

Inheritance diagram for PhotoFinish::D_PNG:



Public Member Functions

- D_PNG ()
- void read_config (const YAML::Node &node)

  Read a D_PNG record from a YAML file.

Additional Inherited Members

6.7.1 Detailed Description

PNG parameters for destination.

Definition at line 139 of file Destination_items.hh.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 PhotoFinish::D_PNG::D_PNG ( )

Definition at line 158 of file Destination_items.cc.

6.7.3 Member Function Documentation

6.7.3.1 void PhotoFinish::D_PNG::read_config (const YAML::Node & *node*)

Read a D_PNG record from a YAML file.

Definition at line 162 of file Destination_items.cc.

The documentation for this class was generated from the following files:

- Destination_items.hh
- Destination_items.cc

6.8 PhotoFinish::D_profile Class Reference

ICC profile parameters for destination.

```
#include <Destination_items.hh>
```

Public Types

- typedef std::shared_ptr $<$ D_profile $>$ ptr

  Shared pointer for a D_profile.

Public Member Functions

- D_profile ()

  Empty constructor.
- D_profile (const std::string &name, fs::path filepath)

  Constructor.
- D_profile (const std::string &name, void $*$data, unsigned int data_size)

  Constructor.
- D_profile (const D_profile &other)

  Copy constructor.
- $\sim$D_profile ()

  Destructor.
- D_profile & operator= (const D_profile &b)

  Assignment operator.
- definable$<$ std::string $>$ name (void) const

  Name of the profile.
- definable$<$ fs::path $>$ filepath (void) const

  File path for reading the profile.
- bool has_data (void) const

  Do we have the profile data instead of a file path?
- CMS::Profile::ptr profile (void) const

  The profile data for LCMS2.
- void $*$ data (void) const

  The profile data.
- unsigned int data_size (void) const

  The size of the profile data.
- void read_config (const YAML::Node &node)

  Read a D_profile record from a YAML file.

6.8.1 Detailed Description

ICC profile parameters for destination.

Definition at line 244 of file Destination_items.hh.

6.8.2 Member Typedef Documentation

6.8.2.1 typedef std::shared_ptr$<$D_profile$>$ PhotoFinish::D_profile::ptr

Shared pointer for a D_profile.

Definition at line 291 of file Destination_items.hh.

6.8.3 Constructor & Destructor Documentation

6.8.3.1 PhotoFinish::D_profile::D_profile ( )

Empty constructor.

Definition at line 360 of file Destination_items.cc.

6.8.3.2 PhotoFinish::D_profile::D_profile (const std::string & *name*, fs::path *filepath*)

Constructor.

Definition at line 364 of file Destination_items.cc.

6.8.3.3 PhotoFinish::D_profile::D_profile (const std::string & *name*, void ∗ *data*, unsigned int *data_size*)

Constructor.

Definition at line 370 of file Destination_items.cc.

6.8.3.4 PhotoFinish::D_profile::D_profile (const D_profile & *other*)

Copy constructor.

Definition at line 375 of file Destination_items.cc.

6.8.3.5 PhotoFinish::D_profile::∼D_profile ( )

Destructor.

Definition at line 382 of file Destination_items.cc.

6.8.4 Member Function Documentation

6.8.4.1 void∗ PhotoFinish::D_profile::data (void) const [inline]

The profile data.

Definition at line 283 of file Destination_items.hh.

6.8.4.2 unsigned int PhotoFinish::D_profile::data_size (void) const [inline]

The size of the profile data.

Definition at line 286 of file Destination_items.hh.

6.8.4.3 definable<fs::path> PhotoFinish::D_profile::filepath (void) const [inline]

File path for reading the profile.

Definition at line 274 of file Destination_items.hh.

6.8.4.4 bool PhotoFinish::D_profile::has_data (void) const [inline]

Do we have the profile data instead of a file path?

Definition at line 277 of file Destination_items.hh.

6.8.4.5 definable< std::string > PhotoFinish::D_profile::name (void) const [inline]

Name of the profile.

Definition at line 271 of file Destination_items.hh.

6.8.4.6 D_profile & PhotoFinish::D_profile::operator= (const D_profile & *b*)

Assignment operator.

Definition at line 390 of file Destination_items.cc.

6.8.4.7 CMS::Profile::ptr PhotoFinish::D_profile::profile (void) const

The profile data for LCMS2.

Definition at line 402 of file Destination_items.cc.

6.8.4.8 void PhotoFinish::D_profile::read_config (const YAML::Node & *node*)

Read a D_profile record from a YAML file.

Definition at line 412 of file Destination_items.cc.

The documentation for this class was generated from the following files:

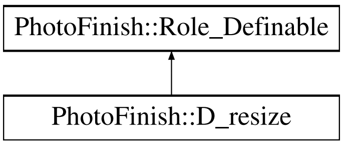
- Destination_items.hh
- Destination_items.cc

6.9 PhotoFinish::D_resize Class Reference

Resize parameters for destination.

`#include <Destination_items.hh>`

Inheritance diagram for PhotoFinish::D_resize:



Public Member Functions

- D_resize ()

  Empty constructor.
- definable< std::string > filter (void) const
- definable< double > support (void) const
- void read_config (const YAML::Node &node)

  Read a D_resize record from a YAML file.

Static Public Member Functions

- static D_resize lanczos (double r)

  Named constructor.

Additional Inherited Members

6.9.1 Detailed Description

Resize parameters for destination.

Definition at line 60 of file Destination_items.hh.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 PhotoFinish::D_resize::D_resize ( )

Empty constructor.

Definition at line 48 of file Destination_items.cc.

6.9.3 Member Function Documentation

6.9.3.1 definable<std::string> PhotoFinish::D_resize::filter (void) const [inline]

Definition at line 77 of file Destination_items.hh.

6.9.3.2 static D_resize PhotoFinish::D_resize::lanczos (double *r*) [inline], [static]

Named constructor.

Constructs a D_resize object with filter="lanczos" and the supplied radius

Parameters

r	Radius of Lanczos filter

Definition at line 75 of file Destination_items.hh.

6.9.3.3 void PhotoFinish::D_resize::read_config (const YAML::Node & *node*)

Read a D_resize record from a YAML file.

Definition at line 56 of file Destination_items.cc.

6.9.3.4 definable<double> PhotoFinish::D_resize::support (void) const [inline]

Definition at line 78 of file Destination_items.hh.

The documentation for this class was generated from the following files:

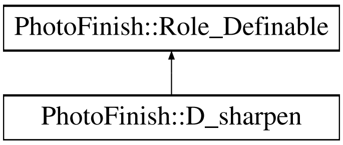
- Destination_items.hh
- Destination_items.cc

6.10 PhotoFinish::D_sharpen Class Reference

Sharpen parameters for destination.

`#include <Destination_items.hh>`

Inheritance diagram for PhotoFinish::D_sharpen:



Public Member Functions

- D_sharpen ()

  Empty constructor.
- definable< double > radius (void) const
- definable< double > sigma (void) const
- void read_config (const YAML::Node &node)

  Read a D_sharpen record from a YAML file.

Additional Inherited Members

6.10.1 Detailed Description

Sharpen parameters for destination.

Definition at line 45 of file Destination_items.hh.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 PhotoFinish::D_sharpen::D_sharpen ( )

Empty constructor.

Definition at line 32 of file Destination_items.cc.

6.10.3 Member Function Documentation

6.10.3.1 **definable**<double> PhotoFinish::D_sharpen::radius (void) const [inline]

Definition at line 53 of file Destination_items.hh.

6.10.3.2 void PhotoFinish::D_sharpen::read_config (const YAML::Node & *node*)

Read a D_sharpen record from a YAML file.

Definition at line 36 of file Destination_items.cc.

6.10.3.3 **definable**<double> PhotoFinish::D_sharpen::sigma (void) const [inline]

Definition at line 54 of file Destination_items.hh.

The documentation for this class was generated from the following files:

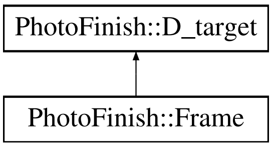
- Destination_items.hh
- Destination_items.cc

6.11 PhotoFinish::D_target Class Reference

Target parameters for destination.

`#include <Destination_items.hh>`

Inheritance diagram for PhotoFinish::D_target:



Public Types

- typedef std::shared_ptr< D_target > ptr

Public Member Functions

- D_target (const std::string &n, double w, double h)

    A target-specific size (in inches) to override the one in the destination.
- D_target (const std::string &n)
- std::string name (void) const
- definable< double > width (void) const
- definable< double > height (void) const
- definable< double > size (void) const
- void read_config (const YAML::Node &node)

    Read a D_target record from a YAML file.

Protected Attributes

- std::string _name
- definable< double > _width
- definable< double > _height
- definable< double > _size

6.11.1 Detailed Description

Target parameters for destination.

Definition at line 84 of file Destination_items.hh.

6.11.2 Member Typedef Documentation

6.11.2.1 typedef std::shared_ptr<D_target> PhotoFinish::D_target::ptr

Definition at line 101 of file Destination_items.hh.

6.11.3 Constructor & Destructor Documentation

6.11.3.1 PhotoFinish::D_target::D_target (const std::string & *n*, double *w*, double *h*)

A target-specific size (in inches) to override the one in the destination.

Definition at line 68 of file Destination_items.cc.

6.11.3.2 PhotoFinish::D_target::D_target (const std::string & *n*)

Definition at line 73 of file Destination_items.cc.

6.11.4 Member Function Documentation

6.11.4.1 definable<double> PhotoFinish::D_target::height (void) const [inline]

Definition at line 96 of file Destination_items.hh.

6.11.4.2 std::string PhotoFinish::D_target::name (void) const [inline]

Definition at line 94 of file Destination_items.hh.

6.11.4.3 void PhotoFinish::D_target::read_config (const YAML::Node & *node*)

Read a D_target record from a YAML file.

Definition at line 78 of file Destination_items.cc.

6.11.4.4 definable<double> PhotoFinish::D_target::size (void) const [inline]

Definition at line 97 of file Destination_items.hh.

6.11.4.5 definable<double> PhotoFinish::D_target::width (void) const [inline]

Definition at line 95 of file Destination_items.hh.

6.11.5 Member Data Documentation

6.11.5.1 definable<double> PhotoFinish::D_target::_height [protected]

Definition at line 87 of file Destination_items.hh.

6.11.5.2 std::string PhotoFinish::D_target::_name [protected]

Definition at line 86 of file Destination_items.hh.

6.11.5.3 definable<double> PhotoFinish::D_target::_size [protected]

Definition at line 88 of file Destination_items.hh.

6.11.5.4 definable< double > PhotoFinish::D_target::_width [protected]

Definition at line 87 of file Destination_items.hh.

The documentation for this class was generated from the following files:

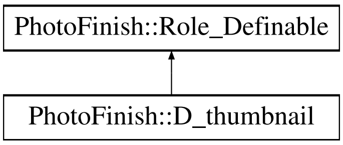
- Destination_items.hh
- Destination_items.cc

6.12 PhotoFinish::D_thumbnail Class Reference

Thumbnail parameters for destination.

`#include <Destination_items.hh>`

Inheritance diagram for PhotoFinish::D_thumbnail:



Public Member Functions

- D_thumbnail ()
- definable< bool > generate (void) const
- definable< double > maxwidth (void) const
- definable< double > maxheight (void) const
- void read_config (const YAML::Node &node)

    Read a D_thumbnail record from a YAML file.

Additional Inherited Members

6.12.1 Detailed Description

Thumbnail parameters for destination.

Definition at line 295 of file Destination_items.hh.

6.12.2 Constructor & Destructor Documentation

6.12.2.1 PhotoFinish::D_thumbnail::D_thumbnail ( )

Definition at line 421 of file Destination_items.cc.

6.12.3 Member Function Documentation

6.12.3.1 definable<bool> PhotoFinish::D_thumbnail::generate (void) const [inline]

Definition at line 303 of file Destination_items.hh.

6.12.3.2 definable< double > PhotoFinish::D_thumbnail::maxheight (void) const [inline]

Definition at line 305 of file Destination_items.hh.

6.12.3.3 definable< double > PhotoFinish::D_thumbnail::maxwidth (void) const [inline]

Definition at line 304 of file Destination_items.hh.

6.12.3.4 void PhotoFinish::D_thumbnail::read_config (const YAML::Node & *node*)

Read a D_thumbnail record from a YAML file.

Definition at line 425 of file Destination_items.cc.

The documentation for this class was generated from the following files:

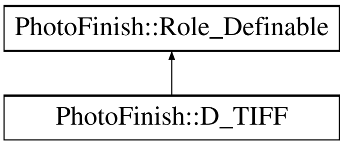
- Destination_items.hh
- Destination_items.cc

6.13 PhotoFinish::D_TIFF Class Reference

TIFF parameters for destination.

`#include <Destination_items.hh>`

Inheritance diagram for PhotoFinish::D_TIFF:



Public Member Functions

- D_TIFF ()

  Empty constructor.
- D_TIFF (const std::string &c)

  Constructor.
- void add_variables (multihash &vars)

  Set values from a map of "variables".
- definable< std::string > artist (void) const
- void set_artist (const std::string &a)
- definable< std::string > copyright (void) const
- void set_copyright (const std::string &c)
- definable< std::string > compression (void) const
- void set_compression (const std::string &c)
- void read_config (const YAML::Node &node)

  Read a D_TIFF record from a YAML file.

Additional Inherited Members

6.13.1 Detailed Description

TIFF parameters for destination.

Definition at line 149 of file Destination_items.hh.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 PhotoFinish::D_TIFF::D_TIFF ( )

Empty constructor.

Definition at line 167 of file Destination_items.cc.

6.13.2.2 PhotoFinish::D_TIFF::D_TIFF (const std::string & *c*)

Constructor.

Parameters

c	Compression string

Definition at line 170 of file Destination_items.cc.

6.13.3 Member Function Documentation

6.13.3.1 void PhotoFinish::D_TIFF::add_variables (multihash & *vars*)

Set values from a map of "variables".

Definition at line 174 of file Destination_items.cc.

6.13.3.2 definable<std::string> PhotoFinish::D_TIFF::artist (void) const [inline]

Definition at line 167 of file Destination_items.hh.

6.13.3.3 definable<std::string> PhotoFinish::D_TIFF::compression (void) const [inline]

Definition at line 173 of file Destination_items.hh.

6.13.3.4 definable<std::string> PhotoFinish::D_TIFF::copyright (void) const [inline]

Definition at line 170 of file Destination_items.hh.

6.13.3.5 void PhotoFinish::D_TIFF::read_config (const YAML::Node & *node*)

Read a D_TIFF record from a YAML file.

Definition at line 202 of file Destination_items.cc.

6.13.3.6 void PhotoFinish::D_TIFF::set_artist (const std::string & *a*) [inline]

Definition at line 168 of file Destination_items.hh.

6.13.3.7 void PhotoFinish::D_TIFF::set_compression (const std::string & *c*) [inline]

Definition at line 174 of file Destination_items.hh.

6.13.3.8 void PhotoFinish::D_TIFF::set_copyright (const std::string & *c*) [inline]

Definition at line 171 of file Destination_items.hh.

The documentation for this class was generated from the following files:

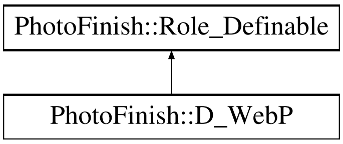
- Destination_items.hh
- Destination_items.cc

6.14 PhotoFinish::D_WebP Class Reference

WebP parameters for destination.

`#include <Destination_items.hh>`

Inheritance diagram for PhotoFinish::D_WebP:



Public Member Functions

- D_WebP ()

  Empty constructor.
- void add_variables (multihash &vars)

  Set values from a map of "variables".
- definable< std::string > preset (void) const
- void set_preset (const std::string &p)
- definable< bool > lossless (void) const
- definable< bool > lossy (void) const
- void set_lossless (bool l=true)
- void set_lossy (bool l=true)
- float quality (void) const
- void set_quality (float q)
- definable< unsigned char > method (void) const
- void set_method (unsigned char m)
- void read_config (const YAML::Node &node)

Additional Inherited Members

6.14.1 Detailed Description

WebP parameters for destination.

Definition at line 212 of file Destination_items.hh.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 PhotoFinish::D_WebP::D_WebP ( )

Empty constructor.

Definition at line 334 of file Destination_items.cc.

6.14.3 Member Function Documentation

6.14.3.1 void PhotoFinish::D_WebP::add_variables (multihash & *vars*)

Set values from a map of "variables".

Definition at line 338 of file Destination_items.cc.

6.14.3.2 definable<bool> PhotoFinish::D_WebP::lossless (void) const [inline]

Definition at line 229 of file Destination_items.hh.

6.14.3.3 definable<bool> PhotoFinish::D_WebP::lossy (void) const [inline]

Definition at line 230 of file Destination_items.hh.

6.14.3.4 definable<unsigned char> PhotoFinish::D_WebP::method (void) const [inline]

Definition at line 237 of file Destination_items.hh.

6.14.3.5 definable<std::string> PhotoFinish::D_WebP::preset (void) const [inline]

Definition at line 226 of file Destination_items.hh.

6.14.3.6 float PhotoFinish::D_WebP::quality (void) const [inline]

Definition at line 234 of file Destination_items.hh.

6.14.3.7 void PhotoFinish::D_WebP::read_config (const YAML::Node & *node*)

Definition at line 341 of file Destination_items.cc.

6.14.3.8 void PhotoFinish::D_WebP::set_lossless (bool *l* = true) [inline]

Definition at line 231 of file Destination_items.hh.

6.14.3.9 void PhotoFinish::D_WebP::set_lossy (bool *l* = true) [inline]

Definition at line 232 of file Destination_items.hh.

6.14.3.10 void PhotoFinish::D_WebP::set_method (unsigned char *m*) [inline]

Definition at line 238 of file Destination_items.hh.

6.14.3.11 void PhotoFinish::D_WebP::set_preset (const std::string & *p*) [inline]

Definition at line 227 of file Destination_items.hh.

6.14.3.12 void PhotoFinish::D_WebP::set_quality (float *q*) [inline]

Definition at line 235 of file Destination_items.hh.

The documentation for this class was generated from the following files:

- Destination_items.hh
- Destination_items.cc

6.15 PhotoFinish::definable< T > Class Template Reference

Template class for storing things that can be defined or undefined.

`#include <Definable.hh>`

Public Member Functions

- definable ()

  Empty constructor.
- definable (const T &i)

  Construct from an item.
- const bool defined (void) const

  Is this object defined?
- void set_defined (bool v=true)

  Set this object as 'defined' (or not)
- void undefine (void)

  Undefine the object.
- T get (void)

  Get the item.
- const T & get (void) const

  Get the item, const version.
- operator T (void) const

  Cast to the contained type.
- T ∗ operator-> ()

  Arrow operator.
- const T ∗ operator-> () const

  Arrow operator, const version.
- definable< T > & operator= (const T &i)

  Assignment operator.

Friends

- std::ostream & operator<< (std::ostream &out, definable< T > &data)

  Allow the contained data to be output to an ostream.

6.15.1 Detailed Description

template<typename T>class PhotoFinish::definable< T >

Template class for storing things that can be defined or undefined.

Definition at line 29 of file Definable.hh.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 template<typename T> PhotoFinish::definable< T >::definable ( ) [inline]

Empty constructor.

This sets the object to undefined and the item is initialised with its empty constructor

Definition at line 39 of file Definable.hh.

6.15.2.2 template<typename T> PhotoFinish::definable< T >::definable (const T & *i*) [inline]

Construct from an item.

This obviously also sets the object to 'defined'

Definition at line 48 of file Definable.hh.

6.15.3 Member Function Documentation

6.15.3.1 template<typename T> const bool PhotoFinish::definable< T >::defined (void) const [inline]

Is this object defined?

Definition at line 54 of file Definable.hh.

6.15.3.2 template<typename T> T PhotoFinish::definable< T >::get (void) [inline]

Get the item.

Definition at line 65 of file Definable.hh.

6.15.3.3 template<typename T> const T& PhotoFinish::definable< T >::get (void) const [inline]

Get the item, const version.

Definition at line 67 of file Definable.hh.

6.15.3.4 template<typename T> PhotoFinish::definable< T >::operator T (void) const [inline]

Cast to the contained type.

Definition at line 70 of file Definable.hh.

6.15.3.5 template<typename T> T* PhotoFinish::definable< T >::operator-> ( ) [inline]

Arrow operator.

Definition at line 73 of file Definable.hh.

6.15.3.6 template<typename T> const T∗ PhotoFinish::definable< T >::operator-> ( ) const [inline]

Arrow operator, const version.

Definition at line 75 of file Definable.hh.

6.15.3.7 template<typename T> definable<T>& PhotoFinish::definable< T >::operator= (const T & *i*) [inline]

Assignment operator.

Definition at line 78 of file Definable.hh.

6.15.3.8 template<typename T> void PhotoFinish::definable< T >::set_defined (bool *v* = true) [inline]

Set this object as 'defined' (or not)

Definition at line 59 of file Definable.hh.

6.15.3.9 template<typename T> void PhotoFinish::definable< T >::undefine (void) [inline]

Undefine the object.

Definition at line 62 of file Definable.hh.

6.15.4 Friends And Related Function Documentation

6.15.4.1 template<typename T> std::ostream& operator<< (std::ostream & *out,* definable< T > & *data*) [friend]

Allow the contained data to be output to an ostream.

Outputs "[undefined]" if the value is not defined.

Definition at line 88 of file Definable.hh.

The documentation for this class was generated from the following file:

- Definable.hh

6.16 PhotoFinish::Destination Class Reference

Represents a destination, read from destinations.yml.

`#include <Destination.hh>`

Public Types

- typedef std::shared_ptr
  < Destination > ptr

  Shared pointer for a Destination.

Public Member Functions

- Destination ()

  Empty constructor.

- Destination (const Destination &other)

    Copy constructor.
- ∼Destination ()

    Destructor.
- Destination & operator= (const Destination &b)

    Assignment operator.
- ptr dupe (void)

    Duplicate.
- ptr add_variables (multihash &vars)

    Duplicate the current object and incorporate variables.
- Frame::ptr best_frame (Image::ptr img)

    Find the best crop+rescaling frame for an image.
- definable< std::string > name (void) const
- const definable< fs::path > & dir (void) const
- definable< double > size (void) const
- const D_sharpen & sharpen (void) const
- const D_resize & resize (void) const
- int num_targets (void) const
- bool has_targets (void) const
- const std::map< std::string, D_target::ptr > & targets (void) const
- definable< std::string > format (void) const
- definable< int > depth (void) const
- void set_depth (int d)
- definable< bool > noresize (void) const
- D_JPEG & jpeg (void)
- void set_jpeg (const D_JPEG &j)
- D_PNG & png (void)
- void set_png (const D_PNG &p)
- D_TIFF & tiff (void)
- void set_tiff (const D_TIFF &t)
- D_JP2 & jp2 (void)
- void set_jp2 (const D_JP2 &j)
- D_WebP & webp (void)
- void set_webp (const D_WebP &w)
- definable< CMS::Intent > intent (void) const
- CMS::Format modify_format (CMS::Format format)

    Modify an LCMS2 pixel format using some of the parameters in the destination.
- CMS::Profile::ptr get_profile (CMS::ColourModel default_colourmodel, std::string for_desc)

    Return an LCMS2 profile object from the profile data.
- const D_profile::ptr profile (void) const
- void set_profile (std::string name, fs::path filepath)
- void set_profile (std::string name, void ∗data, unsigned int data_size)
- void clear_profile (void)
- definable< bool > forcergb (void) const
- definable< bool > forcegrey (void) const
- const D_thumbnail & thumbnail (void) const
- void read_config (const YAML::Node &node)

    Read a destination record from a YAML document.

6.16.1 Detailed Description

Represents a destination, read from destinations.yml.

Definition at line 37 of file Destination.hh.

6.16.2 Member Typedef Documentation

6.16.2.1 typedef std::shared_ptr<Destination> PhotoFinish::Destination::ptr

Shared pointer for a Destination.

Definition at line 86 of file Destination.hh.

6.16.3 Constructor & Destructor Documentation

6.16.3.1 PhotoFinish::Destination::Destination ( )

Empty constructor.

Definition at line 34 of file Destination.cc.

6.16.3.2 PhotoFinish::Destination::Destination (const Destination & *other*)

Copy constructor.

Definition at line 37 of file Destination.cc.

6.16.3.3 PhotoFinish::Destination::∼Destination ( )

Destructor.

Definition at line 57 of file Destination.cc.

6.16.4 Member Function Documentation

6.16.4.1 Destination::ptr PhotoFinish::Destination::add_variables (multihash & *vars*)

Duplicate the current object and incorporate variables.

Definition at line 89 of file Destination.cc.

6.16.4.2 Frame::ptr PhotoFinish::Destination::best_frame (Image::ptr *img*)

Find the best crop+rescaling frame for an image.

Definition at line 98 of file Destination.cc.

6.16.4.3 void PhotoFinish::Destination::clear_profile (void) [inline]

Definition at line 144 of file Destination.hh.

6.16.4.4 definable<int> PhotoFinish::Destination::depth (void) const [inline]

Definition at line 113 of file Destination.hh.

6.16.4.5 const definable<fs::path>& PhotoFinish::Destination::dir (void) const [inline]

Definition at line 99 of file Destination.hh.

6.16.4.6 ptr PhotoFinish::Destination::dupe (void) [inline]

Duplicate.

Definition at line 89 of file Destination.hh.

6.16.4.7 definable<bool> PhotoFinish::Destination::forcegrey (void) const [inline]

Definition at line 147 of file Destination.hh.

6.16.4.8 definable<bool> PhotoFinish::Destination::forcergb (void) const [inline]

Definition at line 146 of file Destination.hh.

6.16.4.9 definable<std::string> PhotoFinish::Destination::format (void) const [inline]

Definition at line 111 of file Destination.hh.

6.16.4.10 CMS::Profile::ptr PhotoFinish::Destination::get_profile (CMS::ColourModel *default_colourmodel,* std::string *for_desc*)

Return an LCMS2 profile object from the profile data.

Definition at line 188 of file Destination.cc.

6.16.4.11 bool PhotoFinish::Destination::has_targets (void) const [inline]

Definition at line 108 of file Destination.hh.

6.16.4.12 definable<CMS::Intent> PhotoFinish::Destination::intent (void) const [inline]

Definition at line 133 of file Destination.hh.

6.16.4.13 D_JP2& PhotoFinish::Destination::jp2 (void) [inline]

Definition at line 127 of file Destination.hh.

6.16.4.14 D_JPEG& PhotoFinish::Destination::jpeg (void) [inline]

Definition at line 118 of file Destination.hh.

6.16.4.15 CMS::Format PhotoFinish::Destination::modify_format (CMS::Format *format*)

Modify an LCMS2 pixel format using some of the parameters in the destination.

Definition at line 152 of file Destination.cc.

6.16.4.16 definable<std::string> PhotoFinish::Destination::name (void) const [inline]

Definition at line 97 of file Destination.hh.

6.16.4.17 definable< bool > PhotoFinish::Destination::noresize (void) const [inline]

Definition at line 116 of file Destination.hh.

6.16.4.18 int PhotoFinish::Destination::num_targets (void) const [inline]

Definition at line 107 of file Destination.hh.

6.16.4.19 Destination & PhotoFinish::Destination::operator= (const Destination & *b*)

Assignment operator.

Definition at line 60 of file Destination.cc.

6.16.4.20 D_PNG& PhotoFinish::Destination::png (void) [inline]

Definition at line 121 of file Destination.hh.

6.16.4.21 const D_profile::ptr PhotoFinish::Destination::profile (void) const [inline]

Definition at line 141 of file Destination.hh.

6.16.4.22 void PhotoFinish::Destination::read_config (const YAML::Node & *node*)

Read a destination record from a YAML document.

Read a Destination record from a YAML file.

Definition at line 205 of file Destination.cc.

6.16.4.23 const D_resize& PhotoFinish::Destination::resize (void) const [inline]

Definition at line 105 of file Destination.hh.

6.16.4.24 void PhotoFinish::Destination::set_depth (int *d*) [inline]

Definition at line 114 of file Destination.hh.

6.16.4.25 void PhotoFinish::Destination::set_jp2 (const D_JP2 & *j*) [inline]

Definition at line 128 of file Destination.hh.

6.16.4.26 void PhotoFinish::Destination::set_jpeg (const D_JPEG & *j*) [inline]

Definition at line 119 of file Destination.hh.

6.16.4.27 void PhotoFinish::Destination::set_png (const D_PNG & *p*) [inline]

Definition at line 122 of file Destination.hh.

6.16.4.28 void PhotoFinish::Destination::set_profile (std::string *name*, fs::path *filepath*) `[inline]`

Definition at line 142 of file Destination.hh.

6.16.4.29 void PhotoFinish::Destination::set_profile (std::string *name*, void ∗ *data*, unsigned int *data_size*) `[inline]`

Definition at line 143 of file Destination.hh.

6.16.4.30 void PhotoFinish::Destination::set_tiff (const D_TIFF & *t*) `[inline]`

Definition at line 125 of file Destination.hh.

6.16.4.31 void PhotoFinish::Destination::set_webp (const D_WebP & *w*) `[inline]`

Definition at line 131 of file Destination.hh.

6.16.4.32 const D_sharpen& PhotoFinish::Destination::sharpen (void) const `[inline]`

Definition at line 103 of file Destination.hh.

6.16.4.33 definable<double> PhotoFinish::Destination::size (void) const `[inline]`

Definition at line 101 of file Destination.hh.

6.16.4.34 const std::map<std::string, D_target::ptr>& PhotoFinish::Destination::targets (void) const `[inline]`

Definition at line 109 of file Destination.hh.

6.16.4.35 const D_thumbnail& PhotoFinish::Destination::thumbnail (void) const `[inline]`

Definition at line 149 of file Destination.hh.

6.16.4.36 D_TIFF& PhotoFinish::Destination::tiff (void) `[inline]`

Definition at line 124 of file Destination.hh.

6.16.4.37 D_WebP& PhotoFinish::Destination::webp (void) `[inline]`

Definition at line 130 of file Destination.hh.

The documentation for this class was generated from the following files:

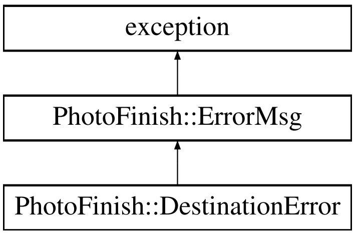
- Destination.hh
- Destination.cc

6.17 PhotoFinish::DestinationError Class Reference

Destination exception.

`#include <Exception.hh>`

Inheritance diagram for PhotoFinish::DestinationError:



Public Member Functions

- DestinationError (const std::string &p, const std::string &v)

  Constructor.
- virtual const char ∗ what () const throw ()

Additional Inherited Members

6.17.1 Detailed Description

Destination exception.

Definition at line 263 of file Exception.hh.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 PhotoFinish::DestinationError::DestinationError (const std::string & *p*, const std::string & *v*) `[inline]`

Constructor.

Parameters

p	Destination field "path"
v	Value that is wrong

Definition at line 273 of file Exception.hh.

6.17.3 Member Function Documentation

6.17.3.1 virtual const char∗ PhotoFinish::DestinationError::what ( ) const throw) `[inline]`, `[virtual]`

Implements PhotoFinish::ErrorMsg.

Definition at line 277 of file Exception.hh.

The documentation for this class was generated from the following file:

- Exception.hh

6.18 PhotoFinish::Destinations Class Reference

A wrapper class for reading destinations from a YAML file and storing them in a map.

```
#include <Destination.hh>
```

Public Types

- typedef std::map< std::string,
  Destination::ptr >::iterator iterator

  Iterator for stepping through destinations.
- typedef std::map< std::string,
  Destination::ptr >
  ::const_iterator const_iterator

  Constant iterator for stepping through destinations.

Public Member Functions

- Destinations (fs::path filepath)
- Destinations (const Destinations &other)
- ∼Destinations ()
- Destinations & operator= (const Destinations &b)
- void Load (fs::path filepath)
- std::map< std::string,
  Destination::ptr >::size_type count (const std::string &key) const
- iterator begin (void)
- const_iterator begin (void) const
- iterator end (void)
- const_iterator end (void) const
- Destination::ptr operator[ ] (const std::string &key)

Friends

- iterator begin (Destinations &d)
- iterator end (Destinations &d)

6.18.1 Detailed Description

A wrapper class for reading destinations from a YAML file and storing them in a map.

Definition at line 156 of file Destination.hh.

6.18.2 Member Typedef Documentation

6.18.2.1 typedef std::map<std::string, Destination::ptr>::const_iterator PhotoFinish::Destinations::const_iterator

Constant iterator for stepping through destinations.

Definition at line 171 of file Destination.hh.

6.18.2.2 typedef std::map<std::string, Destination::ptr>::iterator PhotoFinish::Destinations::iterator

Iterator for stepping through destinations.

Definition at line 168 of file Destination.hh.

6.18.3 Constructor & Destructor Documentation

6.18.3.1 PhotoFinish::Destinations::Destinations (fs::path *filepath*)

Definition at line 286 of file Destination.cc.

6.18.3.2 PhotoFinish::Destinations::Destinations (const Destinations & *other*)

Definition at line 290 of file Destination.cc.

6.18.3.3 PhotoFinish::Destinations::∼Destinations ( )

Definition at line 295 of file Destination.cc.

6.18.4 Member Function Documentation

6.18.4.1 iterator PhotoFinish::Destinations::begin (void) [inline]

Definition at line 177 of file Destination.hh.

6.18.4.2 const_iterator PhotoFinish::Destinations::begin (void) const [inline]

Definition at line 178 of file Destination.hh.

6.18.4.3 std::map<std::string, Destination::ptr>::size_type PhotoFinish::Destinations::count (const std::string & *key*) const [inline]

Definition at line 175 of file Destination.hh.

6.18.4.4 iterator PhotoFinish::Destinations::end (void) [inline]

Definition at line 180 of file Destination.hh.

6.18.4.5 const_iterator PhotoFinish::Destinations::end (void) const [inline]

Definition at line 181 of file Destination.hh.

6.18.4.6 void PhotoFinish::Destinations::Load (fs::path *filepath*)

Definition at line 307 of file Destination.cc.

6.18.4.7 Destinations & PhotoFinish::Destinations::operator= (const Destinations & *b*)

Definition at line 298 of file Destination.cc.

6.18.4.8 Destination::ptr PhotoFinish::Destinations::operator[] (const std::string & *key*) [inline]

Definition at line 189 of file Destination.hh.

6.18.5 Friends And Related Function Documentation

6.18.5.1 iterator begin (**Destinations & *d***) `[friend]`

Definition at line 183 of file Destination.hh.

6.18.5.2 iterator end (**Destinations & *d***) `[friend]`

Definition at line 186 of file Destination.hh.

The documentation for this class was generated from the following files:

- Destination.hh
- Destination.cc

6.19 PhotoFinish::Ditherer Class Reference

Class for dithering images down to 8-bit components.

`#include <Ditherer.hh>`

Public Member Functions

- Ditherer (unsigned int width, unsigned char channels, std::vector< unsigned char > maxvalues={})

  Constructor.
- ∼Ditherer ()

  Destructor.
- void dither (short unsigned int ∗inrow, unsigned char ∗outrow, bool lastrow=false)

  Dither a row of image data.

Static Public Attributes

- static const cmsUInt32Number cmsBaseType = BYTES_SH(2)

  Base LCMS2 base type the ditherer expects the pixels to be in.

6.19.1 Detailed Description

Class for dithering images down to 8-bit components.

Definition at line 27 of file Ditherer.hh.

6.19.2 Constructor & Destructor Documentation

6.19.2.1 PhotoFinish::Ditherer::Ditherer (unsigned int *width*, unsigned char *channels*, std::vector< unsigned char > *maxvalues* = { })

Constructor.

Parameters

width	Width of the image
channels	Number of channels of the image
maxvalues	The maximum values for each channel, defaults to 255 for each

Definition at line 28 of file Ditherer.cc.

6.19.2.2 PhotoFinish::Ditherer::∼Ditherer ( )

Destructor.

Definition at line 51 of file Ditherer.cc.

6.19.3 Member Function Documentation

6.19.3.1 void PhotoFinish::Ditherer::dither (short unsigned int ∗ *inrow*, unsigned char ∗ *outrow*, bool *lastrow* = `false`)

Dither a row of image data.

Performs a Floyd-Steinberg error diffusion dither

Parameters

inrow	Pointer to a row of 16-bit image data
outrow	Pointer to a row 8-bit image data that will be produced
lastrow	Whether this is the last row of the image. Less has to be done.

Definition at line 81 of file Ditherer.cc.

6.19.4 Member Data Documentation

6.19.4.1 const cmsUInt32Number PhotoFinish::Ditherer::cmsBaseType = BYTES_SH(2) `[static]`

Base LCMS2 base type the ditherer expects the pixels to be in.

Users of this class need to add the colour space and number of channels to this base type to be useable.

Definition at line 55 of file Ditherer.hh.

The documentation for this class was generated from the following files:

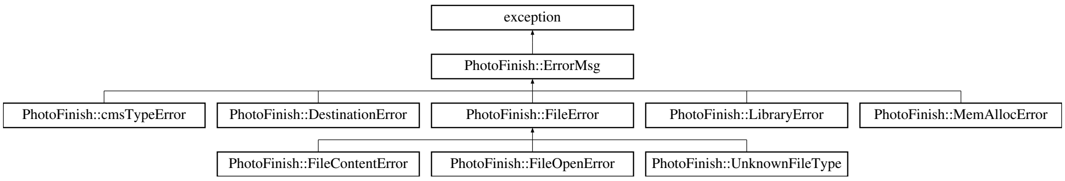
- Ditherer.hh
- Ditherer.cc

6.20 PhotoFinish::ErrorMsg Class Reference

Generic error message exception.

`#include <Exception.hh>`

Inheritance diagram for PhotoFinish::ErrorMsg:


exception
PhotoFinish::ErrorMsg
PhotoFinish::cmsTypeError
PhotoFinish::DestinationError
PhotoFinish::FileError
PhotoFinish::LibraryError
PhotoFinish::MemAllocError
PhotoFinish::FileContentError
PhotoFinish::FileOpenError
PhotoFinish::UnknownFileType

Public Member Functions

- ErrorMsg (const std::string &m)

  Constructor.
- virtual const char ∗ what () const =0 throw ()

Protected Attributes

- const std::string _msg

6.20.1 Detailed Description

Generic error message exception.

Definition at line 117 of file Exception.hh.

6.20.2 Constructor & Destructor Documentation

6.20.2.1 PhotoFinish::ErrorMsg::ErrorMsg (const std::string & *m*) `[inline]`

Constructor.

Parameters

m	Error message

Definition at line 126 of file Exception.hh.

6.20.3 Member Function Documentation

6.20.3.1 virtual const char∗ PhotoFinish::ErrorMsg::what ( ) const throw) `[pure virtual]`

Implemented in PhotoFinish::cmsTypeError, PhotoFinish::LibraryError, PhotoFinish::DestinationError, PhotoFinish::FileContentError, PhotoFinish::FileOpenError, PhotoFinish::UnknownFileType, PhotoFinish::FileError, and PhotoFinish::MemAllocError.

6.20.4 Member Data Documentation

6.20.4.1 const std::string PhotoFinish::ErrorMsg::_msg `[protected]`

Definition at line 119 of file Exception.hh.

The documentation for this class was generated from the following file:

- Exception.hh

6.21 f2vector Union Reference

```
#include <vector.h>
```

Public Attributes

- v2sf v
- float e [2]

6.21.1 Detailed Description

Definition at line 27 of file vector.h.

6.21.2 Member Data Documentation

6.21.2.1 float f2vector::e[2]

Definition at line 30 of file vector.h.

6.21.2.2 v2sf f2vector::v

Definition at line 29 of file vector.h.

The documentation for this union was generated from the following file:

- vector.h

6.22 f4vector Union Reference

```
#include <vector.h>
```

Public Attributes

- v4sf v
- float e [4]

6.22.1 Detailed Description

Definition at line 38 of file vector.h.

6.22.2 Member Data Documentation

6.22.2.1 float f4vector::e[4]

Definition at line 41 of file vector.h.

6.22.2.2 v4sf f4vector::v

Definition at line 40 of file vector.h.

The documentation for this union was generated from the following file:

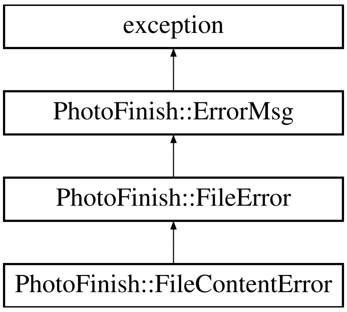
- vector.h

6.23 PhotoFinish::FileContentError Class Reference

File content exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::FileContentError:



Public Member Functions

- FileContentError (const std::string &fp, const std::string &m)

    Constructor.
- FileContentError (const std::string &fp)

    Constructor.
- virtual const char $*$ what () const throw ()

Additional Inherited Members

6.23.1 Detailed Description

File content exception.

Definition at line 234 of file Exception.hh.

6.23.2 Constructor & Destructor Documentation

6.23.2.1 PhotoFinish::FileContentError::FileContentError (const std::string & *fp*, const std::string & *m*) `[inline]`

Constructor.

Parameters

fp	File path
m	Error message

Definition at line 241 of file Exception.hh.

6.23.2.2 PhotoFinish::FileContentError::FileContentError (const std::string & *fp*) `[inline]`

Constructor.

Parameters

fp	File path

Definition at line 249 of file Exception.hh.

6.23.3 Member Function Documentation

6.23.3.1 virtual const char∗ PhotoFinish::FileContentError::what ( ) const throw) [inline], [virtual]

Implements PhotoFinish::FileError.

Definition at line 253 of file Exception.hh.

The documentation for this class was generated from the following file:

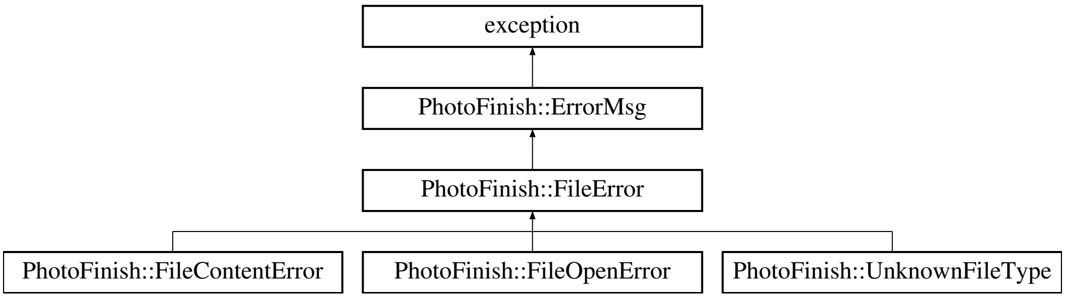
- Exception.hh

6.24 PhotoFinish::FileError Class Reference

File error abstract base exception.

`#include <Exception.hh>`

Inheritance diagram for PhotoFinish::FileError:


exception
PhotoFinish::ErrorMsg
PhotoFinish::FileError
PhotoFinish::FileContentError
PhotoFinish::FileOpenError
PhotoFinish::UnknownFileType

Public Member Functions

- FileError (const std::string &fp, const std::string &m)

  Constructor.
- FileError (const std::string &fp)

  Constructor.
- virtual const char ∗ what () const =0 throw ()

Protected Attributes

- const std::string _filepath

6.24.1 Detailed Description

File error abstract base exception.

Definition at line 150 of file Exception.hh.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 PhotoFinish::FileError::FileError (const std::string & *fp*, const std::string & *m*) [inline]

Constructor.

Parameters

fp	File path
m	Error message

Definition at line 160 of file Exception.hh.

6.24.2.2 PhotoFinish::FileError::FileError (const std::string & *fp*) [inline]

Constructor.

Parameters

fp	File path

Definition at line 168 of file Exception.hh.

6.24.3 Member Function Documentation

6.24.3.1 virtual const char∗ PhotoFinish::FileError::what ( ) const throw) [pure virtual]

Implements PhotoFinish::ErrorMsg.

Implemented in PhotoFinish::FileContentError, PhotoFinish::FileOpenError, and PhotoFinish::UnknownFileType.

6.24.4 Member Data Documentation

6.24.4.1 const std::string PhotoFinish::FileError::_filepath [protected]

Definition at line 152 of file Exception.hh.

The documentation for this class was generated from the following file:

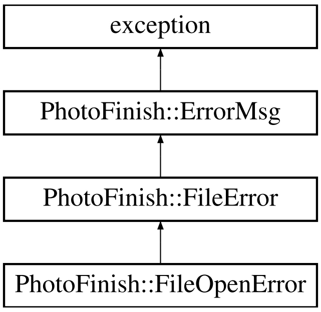
- Exception.hh

6.25 PhotoFinish::FileOpenError Class Reference

File open exception.

`#include <Exception.hh>`

Inheritance diagram for PhotoFinish::FileOpenError:



Public Member Functions

- FileOpenError (const std::string &fp, const std::string &m)

Constructor.

- FileOpenError (const std::string &fp)

  Constructor.

- virtual const char ∗ what () const throw ()

Additional Inherited Members

6.25.1 Detailed Description

File open exception.

Definition at line 205 of file Exception.hh.

6.25.2 Constructor & Destructor Documentation

6.25.2.1 PhotoFinish::FileOpenError::FileOpenError (const std::string & *fp*, const std::string & *m*) [inline]

Constructor.

Parameters

fp	File path
m	Error message

Definition at line 212 of file Exception.hh.

6.25.2.2 PhotoFinish::FileOpenError::FileOpenError (const std::string & *fp*) [inline]

Constructor.

Parameters

fp	File path

Definition at line 220 of file Exception.hh.

6.25.3 Member Function Documentation

6.25.3.1 virtual const char∗ PhotoFinish::FileOpenError::what ( ) const throw) [inline], [virtual]

Implements PhotoFinish::FileError.

Definition at line 224 of file Exception.hh.

The documentation for this class was generated from the following file:

- Exception.hh

6.26 CMS::Format Class Reference

Wrap LCMS2's pixel format.

```
#include <CMS.hh>
```

Public Member Functions

- Format ()

  Empty constructor.
- operator cmsUInt32Number () const

  Cast to an unsigned int for direct use with LCMS2.
- Format & set_8bit (void)

  Set to 8 bit bytes per channel.
- bool is_8bit (void) const

  Is the format 8-bits per channel?
- Format & set_16bit (void)

  Set to 16 bits per channel.
- bool is_16bit (void) const

  Is the format 16-bits (integer) per channel?
- Format & set_32bit (void)

  Set to 32 bits per channel.
- bool is_32bit (void) const

  Is the format 32-bits (integer) per channel?
- Format & set_half (void)

  Set to 16 bit half-precision floating point values per channel.
- bool is_half (void) const

  Is the format a half-precision floating point value(s) per channel?
- Format & set_float (void)

  Set to 32 bit single-precision floating point values per channel.
- bool is_float (void) const

  Is the format a single-precision floating point value(s) per channel?
- Format & set_double (void)

  Set to 64 bit double-precision floating point value(s) per channel.
- bool is_double (void) const

  Is the format a double-precision floating point value(s) per channel?
- Format & set_channel_type (unsigned char bytes, bool fp=false)

  Set the channel type (bytes and float flag)
- Format & set_channel_type (const Format &other)

  Set the channel type (bytes and float flag) from another Format object.
- template<typename P >
  Format & set_channel_type (void)

  Set the channel type (bytes and float flag) from the template type.
- bool is_integer (void) const

  Is the format integer?
- bool is_fp (void) const

  Is the format floating point?
- bool is_optimised (void) const
- unsigned int channels (void) const

  Get the number of channels.
- Format & set_extra_channels (unsigned int e)

  Set the number of 'extra' channels e.g alpha.
- unsigned int extra_channels (void) const

  Get the number of 'extra' channels e.g alpha.
- unsigned int total_channels (void) const

  Get the total number of channels i.e channels() + extra_channels()
- unsigned int bytes_per_channel (void) const

- unsigned int bytes_per_pixel (void) const
- Format & set_swap (bool s=true)

  Set the format as being swapped e.g BGR.
- Format & unset_swap (void)

  Set the format as not being swapped e.g RGB.
- bool is_swapped (void) const

  Is the channel order swapped?
- Format & set_endianswap (bool e=true)
- Format & unset_endianswap (void)
- bool is_endianswapped (void) const
- Format & set_swapfirst (bool f=true)
- Format & unset_swapfirst (void)
- bool is_swappedfirst (void) const
- Format & set_planar (bool p=true)

  Set the format to be planar.
- Format & set_packed (void)

  Set the format to be packed.
- bool is_planar (void) const

  Is the format planar?
- bool is_packed (void) const

  Is the format packed?
- Format & set_vanilla (bool v=true)

  Set the flavour to 'vanilla' i.e minimum value is white.
- Format & set_chocolate (void)

  Set the flavour to 'chocolate' i.e minimum value is black.
- bool is_vanilla (void) const

  Is the flavour 'vanilla'? i.e minimum value is white.
- bool is_chocolate (void) const

  Is the flavour 'chocolate'? i.e minimum value is black.
- Format & set_colour_model (const ColourModel cm, unsigned int channels=0)
- ColourModel colour_model (void) const

  Get the colour model of the pixel format.
- Format & set_premult_alpha (bool pa=true)
- Format & unset_premult_alpha ()
- bool is_premult_alpha (void) const
- template<typename T >
  T scaleval (void)

  Get the maximum value used/supported by this format.
- template<>
  Format & set_channel_type (void)
- template<>
  Format & set_channel_type (void)
- template<>
  Format & set_channel_type (void)
- template<>
  Format & set_channel_type (void)
- template<>
  Format & set_channel_type (void)

Static Public Member Functions

- static Format Grey8 (void)

  Named constructor.
- static Format Grey16 (void)

  Named constructor.
- static Format RGB8 (void)

  Named constructor.
- static Format RGB16 (void)

  Named constructor.
- static Format CMYK8 (void)

  Named constructor.
- static Format LabFloat (void)

  Named constructor.
- static Format LabDouble (void)

  Named constructor.

Friends

- class Transform

6.26.1 Detailed Description

Wrap LCMS2's pixel format.

Definition at line 131 of file CMS.hh.

6.26.2 Constructor & Destructor Documentation

6.26.2.1 CMS::Format::Format ( )

Empty constructor.

Definition at line 205 of file CMS.cc.

6.26.3 Member Function Documentation

6.26.3.1 unsigned int CMS::Format::bytes_per_channel (void) const [inline]

Definition at line 235 of file CMS.hh.

6.26.3.2 unsigned int CMS::Format::bytes_per_pixel (void) const [inline]

Definition at line 237 of file CMS.hh.

6.26.3.3 unsigned int CMS::Format::channels (void) const [inline]

Get the number of channels.

Definition at line 224 of file CMS.hh.

6.26.3.4 Format CMS::Format::CMYK8 (void) [static]

Named constructor.

Definition at line 227 of file CMS.cc.

6.26.3.5 ColourModel CMS::Format::colour_model (void) const [inline]

Get the colour model of the pixel format.

Definition at line 290 of file CMS.hh.

6.26.3.6 unsigned int CMS::Format::extra_channels (void) const [inline]

Get the number of 'extra' channels e.g alpha.

Definition at line 230 of file CMS.hh.

6.26.3.7 Format CMS::Format::Grey16 (void) [static]

Named constructor.

Definition at line 215 of file CMS.cc.

6.26.3.8 Format CMS::Format::Grey8 (void) [static]

Named constructor.

Definition at line 211 of file CMS.cc.

6.26.3.9 bool CMS::Format::is_16bit (void) const [inline]

Is the format 16-bits (integer) per channel?

Definition at line 179 of file CMS.hh.

6.26.3.10 bool CMS::Format::is_32bit (void) const [inline]

Is the format 32-bits (integer) per channel?

Definition at line 185 of file CMS.hh.

6.26.3.11 bool CMS::Format::is_8bit (void) const [inline]

Is the format 8-bits per channel?

Definition at line 173 of file CMS.hh.

6.26.3.12 bool CMS::Format::is_chocolate (void) const [inline]

Is the flavour 'chocolate'? i.e minimum value is black.

Definition at line 283 of file CMS.hh.

6.26.3.13 bool CMS::Format::is_double (void) const [inline]

Is the format a double-precision floating point value(s) per channel?

Definition at line 203 of file CMS.hh.

6.26.3.14 bool CMS::Format::is_endianswapped (void) const [inline]

Definition at line 253 of file CMS.hh.

6.26.3.15 bool CMS::Format::is_float (void) const [inline]

Is the format a single-precision floating point value(s) per channel?

Definition at line 197 of file CMS.hh.

6.26.3.16 bool CMS::Format::is_fp (void) const [inline]

Is the format floating point?

Definition at line 219 of file CMS.hh.

6.26.3.17 bool CMS::Format::is_half (void) const [inline]

Is the format a half-precision floating point value(s) per channel?

Definition at line 191 of file CMS.hh.

6.26.3.18 bool CMS::Format::is_integer (void) const [inline]

Is the format integer?

Definition at line 216 of file CMS.hh.

6.26.3.19 bool CMS::Format::is_optimised (void) const [inline]

Definition at line 221 of file CMS.hh.

6.26.3.20 bool CMS::Format::is_packed (void) const [inline]

Is the format packed?

Definition at line 271 of file CMS.hh.

6.26.3.21 bool CMS::Format::is_planar (void) const [inline]

Is the format planar?

Definition at line 268 of file CMS.hh.

6.26.3.22 bool CMS::Format::is_premult_alpha (void) const [inline]

Definition at line 296 of file CMS.hh.

6.26.3.23 bool CMS::Format::is_swapped (void) const [inline]

Is the channel order swapped?

Definition at line 246 of file CMS.hh.

6.26.3.24 bool CMS::Format::is_swappedfirst (void) const [inline]

Definition at line 259 of file CMS.hh.

6.26.3.25 bool CMS::Format::is_vanilla (void) const [inline]

Is the flavour 'vanilla'? i.e minimum value is white.

Definition at line 280 of file CMS.hh.

6.26.3.26 Format CMS::Format::LabDouble (void) [static]

Named constructor.

Definition at line 235 of file CMS.cc.

6.26.3.27 Format CMS::Format::LabFloat (void) [static]

Named constructor.

Definition at line 231 of file CMS.cc.

6.26.3.28 CMS::Format::operator cmsUInt32Number ( ) const [inline]

Cast to an unsigned int for direct use with LCMS2.

Definition at line 146 of file CMS.hh.

6.26.3.29 Format CMS::Format::RGB16 (void) [static]

Named constructor.

Definition at line 223 of file CMS.cc.

6.26.3.30 Format CMS::Format::RGB8 (void) [static]

Named constructor.

Definition at line 219 of file CMS.cc.

6.26.3.31 template<typename T > T CMS::Format::scaleval (void) [inline]

Get the maximum value used/supported by this format.

Definition at line 300 of file CMS.hh.

6.26.3.32 Format & CMS::Format::set_16bit (void)

Set to 16 bits per channel.

Definition at line 259 of file CMS.cc.

6.26.3.33 Format & CMS::Format::set_32bit (void)

Set to 32 bits per channel.

Definition at line 266 of file CMS.cc.

6.26.3.34 Format & CMS::Format::set_8bit (void)

Set to 8 bit bytes per channel.

Definition at line 252 of file CMS.cc.

6.26.3.35 Format & CMS::Format::set_channel_type (unsigned char *bytes*, bool *fp* = `false`)

Set the channel type (bytes and float flag)

Definition at line 297 of file CMS.cc.

6.26.3.36 Format & CMS::Format::set_channel_type (const Format & *other*)

Set the channel type (bytes and float flag) from another Format object.

Definition at line 305 of file CMS.cc.

6.26.3.37 template<typename P > Format& CMS::Format::set_channel_type (void)

Set the channel type (bytes and float flag) from the template type.

6.26.3.38 template<> Format& CMS::Format::set_channel_type (void) `[inline]`

Definition at line 316 of file CMS.hh.

6.26.3.39 template<> Format& CMS::Format::set_channel_type (void) `[inline]`

Definition at line 319 of file CMS.hh.

6.26.3.40 template<> Format& CMS::Format::set_channel_type (void) `[inline]`

Definition at line 322 of file CMS.hh.

6.26.3.41 template<> Format& CMS::Format::set_channel_type (void) `[inline]`

Definition at line 325 of file CMS.hh.

6.26.3.42 template<> Format& CMS::Format::set_channel_type (void) `[inline]`

Definition at line 328 of file CMS.hh.

6.26.3.43 Format & CMS::Format::set_chocolate (void)

Set the flavour to 'chocolate' i.e minimum value is black.

Definition at line 369 of file CMS.cc.

6.26.3.44 Format & CMS::Format::set_colour_model (const ColourModel *cm,* unsigned int *channels =* `0`)

Set the colour model and number of channels 'channels' is only used if the colour model is unknown

Definition at line 374 of file CMS.cc.

6.26.3.45 Format & CMS::Format::set_double (void)

Set to 64 bit double-precision floating point value(s) per channel.

Definition at line 289 of file CMS.cc.

6.26.3.46 Format & CMS::Format::set_endianswap (bool *e =* `true`)

Definition at line 330 of file CMS.cc.

6.26.3.47 Format & CMS::Format::set_extra_channels (unsigned int *e*)

Set the number of 'extra' channels e.g alpha.

Definition at line 313 of file CMS.cc.

6.26.3.48 Format & CMS::Format::set_float (void)

Set to 32 bit single-precision floating point values per channel.

Definition at line 281 of file CMS.cc.

6.26.3.49 Format & CMS::Format::set_half (void)

Set to 16 bit half-precision floating point values per channel.

Definition at line 273 of file CMS.cc.

6.26.3.50 Format & CMS::Format::set_packed (void)

Set the format to be packed.

Definition at line 358 of file CMS.cc.

6.26.3.51 Format & CMS::Format::set_planar (bool *p =* `true`)

Set the format to be planar.

Definition at line 352 of file CMS.cc.

6.26.3.52 Format & CMS::Format::set_premult_alpha (bool *pa =* `true`)

Definition at line 417 of file CMS.cc.

6.26.3.53 Format & CMS::Format::set_swap (bool *s* = true)

Set the format as being swapped e.g BGR.

Definition at line 319 of file CMS.cc.

6.26.3.54 Format & CMS::Format::set_swapfirst (bool *f* = true)

Definition at line 341 of file CMS.cc.

6.26.3.55 Format & CMS::Format::set_vanilla (bool *v* = true)

Set the flavour to 'vanilla' i.e minimum value is white.

Definition at line 363 of file CMS.cc.

6.26.3.56 unsigned int CMS::Format::total_channels (void) const [inline]

Get the total number of channels i.e channels() + extra_channels()

Definition at line 233 of file CMS.hh.

6.26.3.57 Format & CMS::Format::unset_endianswap (void)

Definition at line 336 of file CMS.cc.

6.26.3.58 Format & CMS::Format::unset_premult_alpha ( )

Definition at line 422 of file CMS.cc.

6.26.3.59 Format & CMS::Format::unset_swap (void)

Set the format as not being swapped e.g RGB.

Definition at line 325 of file CMS.cc.

6.26.3.60 Format & CMS::Format::unset_swapfirst (void)

Definition at line 347 of file CMS.cc.

6.26.4 Friends And Related Function Documentation

6.26.4.1 friend class Transform [friend]

Definition at line 139 of file CMS.hh.

The documentation for this class was generated from the following files:

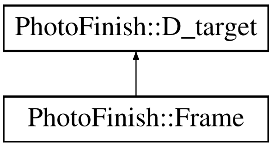
- CMS.hh
- CMS.cc

6.27 PhotoFinish::Frame Class Reference

Crop+rescaling parameters.

`#include <Frame.hh>`

Inheritance diagram for PhotoFinish::Frame:



Public Types

- typedef std::shared_ptr< Frame > ptr

  Shared pointer for a Frame.

Public Member Functions

- Frame (double tw, double th, double x, double y, double w, double h)

  Constructor.
- Frame (const D_target &target, double x, double y, double w, double h)

  Constructor.
- Image::ptr crop_resize (Image::ptr img, const D_resize &dr, bool can_free=false)

  Crop and resize an image.
- const double crop_x (void) const

  The left-most border of the crop window.
- const double crop_y (void) const

  The top-most border of the crop window.
- const double crop_w (void) const

  The width of the crop window.
- const double crop_h (void) const

  The height of the crop window.
- const double waste (Image::ptr img) const

  How much of the original image is wasted by this crop frame.

Additional Inherited Members

6.27.1 Detailed Description

Crop+rescaling parameters.

Definition at line 28 of file Frame.hh.

6.27.2 Member Typedef Documentation

6.27.2.1 typedef std::shared_ptr<Frame> PhotoFinish::Frame::ptr

Shared pointer for a Frame.

Definition at line 71 of file Frame.hh.

6.27.3 Constructor & Destructor Documentation

6.27.3.1 PhotoFinish::Frame::Frame (double *tw*, double *th*, double *x*, double *y*, double *w*, double *h*)

Constructor.

Parameters

tw,th	Size (width, height) of the output
x,y	Top-left corner of crop+rescale window
w,h	Size of the crop+rescale window

Definition at line 29 of file Frame.cc.

6.27.3.2 PhotoFinish::Frame::Frame (const D_target & *target*, double *x*, double *y*, double *w*, double *h*)

Constructor.

Parameters

target	D_target object providing the size (width, height) of the output
x,y	Top-left corner of crop+rescale window
w,h	Size of the crop+rescale window

Definition at line 35 of file Frame.cc.

6.27.4 Member Function Documentation

6.27.4.1 const double PhotoFinish::Frame::crop_h (void) const [inline]

The height of the crop window.

Definition at line 65 of file Frame.hh.

6.27.4.2 Image::ptr PhotoFinish::Frame::crop_resize (Image::ptr *img*, const D_resize & *dr*, bool *can_free* = false)

Crop and resize an image.

Parameters

img	The source image
dr	A D_resize object which will supply our parameters.
can_free	Can each row of the image be freed after it is convolved?

Returns

A new cropped and resized image

Definition at line 41 of file Frame.cc.

6.27.4.3 const double PhotoFinish::Frame::crop_w (void) const [inline]

The width of the crop window.

Definition at line 63 of file Frame.hh.

6.27.4.4 const double PhotoFinish::Frame::crop_x (void) const [inline]

The left-most border of the crop window.

Definition at line 59 of file Frame.hh.

6.27.4.5 const double PhotoFinish::Frame::crop_y (void) const [inline]

The top-most border of the crop window.

Definition at line 61 of file Frame.hh.

6.27.4.6 const double PhotoFinish::Frame::waste (Image::ptr *img*) const

How much of the original image is wasted by this crop frame.

Definition at line 58 of file Frame.cc.

The documentation for this class was generated from the following files:

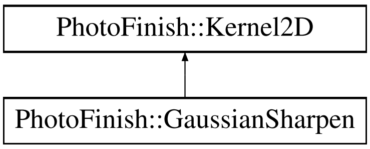
- Frame.hh
- Frame.cc

6.28 PhotoFinish::GaussianSharpen Class Reference

GaussianSharpen kernel.

`#include <Kernel2D.hh>`

Inheritance diagram for PhotoFinish::GaussianSharpen:



Public Member Functions

- GaussianSharpen ()

  Empty constructor.
- GaussianSharpen (const D_sharpen &ds)

  Constructor.

Additional Inherited Members

6.28.1 Detailed Description

GaussianSharpen kernel.

Definition at line 76 of file Kernel2D.hh.

6.28.2 Constructor & Destructor Documentation

6.28.2.1 PhotoFinish::GaussianSharpen::GaussianSharpen ( )

Empty constructor.

Definition at line 247 of file Kernel2D.cc.

6.28.2.2 PhotoFinish::GaussianSharpen::GaussianSharpen (const **D_sharpen** & *ds*)

Constructor.

Parameters

ds	A D_sharpen object which will supply our parameters.

Definition at line 250 of file Kernel2D.cc.

The documentation for this class was generated from the following files:

- Kernel2D.hh
- Kernel2D.cc

6.29 PhotoFinish::Image Class Reference

An image class.

```
#include <Image.hh>
```

Public Types

- typedef std::shared_ptr< Image > ptr

  Shared pointer for an Image.

Public Member Functions

- Image (unsigned int w, unsigned int h, CMS::Format f)

  Constructor.
- ∼Image ()

  Destructor.
- const unsigned int width (void) const

  The width of this image.
- const unsigned int height (void) const

  The height of this image.
- bool has_profile (void) const
- const CMS::Profile::ptr profile (void) const

  Get the ICC profile.
- void set_profile (CMS::Profile::ptr p)

  Set the ICC profile.
- CMS::Format format (void) const

  Get the CMS format.
- const definable< double > xres (void) const

  The X resolution of this image (PPI)
- const definable< double > yres (void) const

  The Y resolution of this image (PPI)
- void set_resolution (double r)

  Set both the X and Y resolution (PPI)
- void set_xres (double r)

  Set the X resolution (PPI)
- void set_yres (double r)

  Set the Y resolution (PPI)

- void set_resolution (double xr, double yr)

  Set the X and Y resolutions (PPI)
- void set_resolution_from_size (double size)

  Set the resolution given the length of the longest side (in inches)
- size_t pixel_size (void) const

  Return the size of a pixel in bytes.
- size_t row_size (void) const

  Retun the size of a row in bytes.
- void check_rowdata_alloc (unsigned int y)
- template<typename T = void>
  T ∗ row (unsigned int y) const

  Pointer to pixel data at start of row.
- template<typename T >
  T ∗ at (unsigned int x, unsigned int y) const

  Pointer to pixel data at coordinates.
- template<typename T >
  T & at (unsigned int x, unsigned int y, unsigned char c) const
- void free_row (unsigned int y)

  Free the memory storing row 'y'.
- Exiv2::ExifData & EXIFtags (void)

  The Exiv2::ExifData object.
- Exiv2::IptcData & IPTCtags (void)

  The Exiv2::IptcData object.
- Exiv2::XmpData & XMPtags (void)

  The Exiv2::XmpData object.
- ptr transform_colour (CMS::Profile::ptr dest_profile, CMS::Format dest_format, CMS::Intent intent=CMS::Intent::Perceptual, bool can_free=false)

  Transform this image into a different colour space and/or ICC profile, making a new image.
- void transform_colour_inplace (CMS::Profile::ptr dest_profile, CMS::Format dest_format, CMS::Intent intent=CMS::Intent::Perceptual)

  Transform this image in-place into a different colour space and/or ICC profile.
- void un_alpha_mult (void)

  Un-pre-multiply the colour values with the alpha channel.
- void alpha_mult (CMS::Format dest_format)

  Pre-multiply the colour values with the alpha.

Static Public Member Functions

- static CMS::Profile::ptr default_profile (CMS::ColourModel default_colourmodel, std::string for_desc)

  Create either an sRGB or greyscale profile depending on image format.
- static CMS::Profile::ptr default_profile (CMS::Format format, std::string for_desc)

6.29.1 Detailed Description

An image class.

Definition at line 31 of file Image.hh.

6.29.2 Member Typedef Documentation

6.29.2.1 typedef std::shared_ptr<Image> PhotoFinish::Image::ptr

Shared pointer for an Image.

Definition at line 55 of file Image.hh.

6.29.3 Constructor & Destructor Documentation

6.29.3.1 PhotoFinish::Image::Image (unsigned int *w*, unsigned int *h*, CMS::Format *f*)

Constructor.

Parameters

w,h	Width and height of the image
t	LCMS2 pixel format

Definition at line 28 of file Image.cc.

6.29.3.2 PhotoFinish::Image::∼Image ( )

Destructor.

Definition at line 44 of file Image.cc.

6.29.4 Member Function Documentation

6.29.4.1 void PhotoFinish::Image::alpha_mult (CMS::Format *dest_format*)

Pre-multiply the colour values with the alpha.

Parameters

dest_format	Destination format, only the channel type (bytes and float flag) are used.

Definition at line 404 of file Image.cc.

6.29.4.2 template<typename T > T∗ PhotoFinish::Image::at (unsigned int *x*, unsigned int *y*) const `[inline]`

Pointer to pixel data at coordinates.

Definition at line 122 of file Image.hh.

6.29.4.3 template<typename T > T& PhotoFinish::Image::at (unsigned int *x*, unsigned int *y*, unsigned char *c*) const `[inline]`

Definition at line 125 of file Image.hh.

6.29.4.4 void PhotoFinish::Image::check_rowdata_alloc (unsigned int *y*) `[inline]`

Definition at line 111 of file Image.hh.

6.29.4.5 CMS::Profile::ptr PhotoFinish::Image::default_profile (CMS::ColourModel *default_colourmodel*, std::string *for_desc*) [static]

Create either an sRGB or greyscale profile depending on image format.

Definition at line 56 of file Image.cc.

6.29.4.6 static CMS::Profile::ptr PhotoFinish::Image::default_profile (CMS::Format *format*, std::string *for_desc*) [inline], [static]

Definition at line 147 of file Image.hh.

6.29.4.7 Exiv2::ExifData& PhotoFinish::Image::EXIFtags (void) [inline]

The Exiv2::ExifData object.

Definition at line 136 of file Image.hh.

6.29.4.8 CMS::Format PhotoFinish::Image::format (void) const [inline]

Get the CMS format.

Definition at line 82 of file Image.hh.

6.29.4.9 void PhotoFinish::Image::free_row (unsigned int *y*) [inline]

Free the memory storing row 'y'.

Definition at line 128 of file Image.hh.

6.29.4.10 bool PhotoFinish::Image::has_profile (void) const [inline]

Definition at line 73 of file Image.hh.

6.29.4.11 const unsigned int PhotoFinish::Image::height (void) const [inline]

The height of this image.

Definition at line 71 of file Image.hh.

6.29.4.12 Exiv2::IptcData& PhotoFinish::Image::IPTCtags (void) [inline]

The Exiv2::IptcData object.

Definition at line 139 of file Image.hh.

6.29.4.13 size_t PhotoFinish::Image::pixel_size (void) const [inline]

Return the size of a pixel in bytes.

Definition at line 106 of file Image.hh.

6.29.4.14 const CMS::Profile::ptr PhotoFinish::Image::profile (void) const [inline]

Get the ICC profile.

Definition at line 76 of file Image.hh.

6.29.4.15 template<typename T = void> T* PhotoFinish::Image::row (unsigned int *y*) const [inline]

Pointer to pixel data at start of row.

Definition at line 118 of file Image.hh.

6.29.4.16 size_t PhotoFinish::Image::row_size (void) const [inline]

Retun the size of a row in bytes.

Definition at line 109 of file Image.hh.

6.29.4.17 void PhotoFinish::Image::set_profile (CMS::Profile::ptr *p*) [inline]

Set the ICC profile.

Definition at line 79 of file Image.hh.

6.29.4.18 void PhotoFinish::Image::set_resolution (double *r*) [inline]

Set both the X and Y resolution (PPI)

Definition at line 91 of file Image.hh.

6.29.4.19 void PhotoFinish::Image::set_resolution (double *xr,* double *yr*) [inline]

Set the X and Y resolutions (PPI)

Definition at line 100 of file Image.hh.

6.29.4.20 void PhotoFinish::Image::set_resolution_from_size (double *size*) [inline]

Set the resolution given the length of the longest side (in inches)

Definition at line 103 of file Image.hh.

6.29.4.21 void PhotoFinish::Image::set_xres (double *r*) [inline]

Set the X resolution (PPI)

Definition at line 94 of file Image.hh.

6.29.4.22 void PhotoFinish::Image::set_yres (double *r*) [inline]

Set the Y resolution (PPI)

Definition at line 97 of file Image.hh.

6.29.4.23 Image::ptr PhotoFinish::Image::transform_colour (CMS::Profile::ptr *dest_profile*, CMS::Format *dest_format*, CMS::Intent *intent* = CMS::Intent::Perceptual, bool *can_free* = `false`)

Transform this image into a different colour space and/or ICC profile, making a new image.

Parameters

dest_profile	The ICC profile of the destination. If NULL, uses image's profile.
dest_format	The LCMS2 pixel format.
intent	The ICC intent of the transform, defaults to perceptual.
can_free	Whether rows can be freed after transforming, defaults to false.

Returns

A new image

Definition at line 146 of file Image.cc.

6.29.4.24 void PhotoFinish::Image::transform_colour_inplace (CMS::Profile::ptr *dest_profile,* CMS::Format *dest_format,* CMS::Intent *intent =* CMS::Intent::Perceptual)

Transform this image in-place into a different colour space and/or ICC profile.

Parameters

dest_profile	The ICC profile of the destination. If NULL, uses image's profile.
dest_format	The LCMS2 pixel format.
intent	The ICC intent of the transform, defaults to perceptual.

Definition at line 204 of file Image.cc.

6.29.4.25 void PhotoFinish::Image::un_alpha_mult (void)

Un-pre-multiply the colour values with the alpha channel.

Converts data to floating point (SAMPLE) in the process

Definition at line 317 of file Image.cc.

6.29.4.26 const unsigned int PhotoFinish::Image::width (void) const [inline]

The width of this image.

Definition at line 68 of file Image.hh.

6.29.4.27 Exiv2::XmpData& PhotoFinish::Image::XMPtags (void) [inline]

The Exiv2::XmpData object.

Definition at line 142 of file Image.hh.

6.29.4.28 const definable<double> PhotoFinish::Image::xres (void) const [inline]

The X resolution of this image (PPI)

Definition at line 85 of file Image.hh.

6.29.4.29 const definable<double> PhotoFinish::Image::yres (void) const [inline]

The Y resolution of this image (PPI)

Definition at line 88 of file Image.hh.

The documentation for this class was generated from the following files:

- Image.hh
- Image.cc

6.30 PhotoFinish::ImageFilepath Class Reference

Class for holding filename and the image format.

```
#include <ImageFile.hh>
```

Public Member Functions

- ImageFilepath (const fs::path filepath, const std::string format)

  Constructor.
- ImageFilepath (const fs::path filepath) throw (UnknownFileType)

  Constructor.
- fs::path fixed_filepath (void) const throw (UnknownFileType)
- void fix_filepath (void) throw (UnknownFileType)
- virtual const fs::path filepath (void) const

  File path of this image file.
- virtual std::string format (void) const

  Format of this image file.

Friends

- std::ostream & operator<< (std::ostream &out, const ImageFilepath &fp)

6.30.1 Detailed Description

Class for holding filename and the image format.

Definition at line 55 of file ImageFile.hh.

6.30.2 Constructor & Destructor Documentation

6.30.2.1 PhotoFinish::ImageFilepath::ImageFilepath (const fs::path *filepath,* const std::string *format*)

Constructor.

Parameters

filepath	The path of the image file
format	Format of the image file

Definition at line 28 of file ImageFile.cc.

6.30.2.2 PhotoFinish::ImageFilepath::ImageFilepath (const fs::path *filepath*) throw UnknownFileType)

Constructor.

Guess the format from the file extension.

Parameters

filepath	The path of the image file

Definition at line 33 of file ImageFile.cc.

6.30.3 Member Function Documentation

6.30.3.1 virtual const fs::path PhotoFinish::ImageFilepath::filepath (void) const [inline], [virtual]

File path of this image file.

Definition at line 80 of file ImageFile.hh.

6.30.3.2 void PhotoFinish::ImageFilepath::fix_filepath (void) throw UnknownFileType) [inline]

Definition at line 77 of file ImageFile.hh.

6.30.3.3 fs::path PhotoFinish::ImageFilepath::fixed_filepath (void) const throw UnknownFileType)

Definition at line 77 of file ImageFile.cc.

6.30.3.4 virtual std::string PhotoFinish::ImageFilepath::format (void) const [inline], [virtual]

Format of this image file.

Definition at line 83 of file ImageFile.hh.

6.30.4 Friends And Related Function Documentation

6.30.4.1 std::ostream& operator<< (std::ostream & *out*, const ImageFilepath & *fp*) [friend]

Definition at line 85 of file ImageFile.hh.

The documentation for this class was generated from the following files:

- ImageFile.hh
- ImageFile.cc

6.31 PhotoFinish::ImageReader Class Reference

Abstract base class for reading image files.

```
#include <ImageFile.hh>
```

Public Types

- typedef std::shared_ptr
  < ImageReader > ptr

  Shared pointer for an ImageReader.

Public Member Functions

- virtual Image::ptr read (void)

  Read the file into an image.
- virtual Image::ptr read (Destination::ptr dest)=0

  Read the file into an image.

Static Public Member Functions

- static ImageReader::ptr open (const ImageFilepath &ifp) throw (UnknownFileType)

  Named constructor.

Protected Member Functions

- ImageReader (const fs::path fp)

  Private constructor.
- void extract_tags (Image::ptr img)

  Extract tags from file.

Protected Attributes

- const fs::path _filepath
- bool _is_open

6.31.1 Detailed Description

Abstract base class for reading image files.

Definition at line 96 of file ImageFile.hh.

6.31.2 Member Typedef Documentation

6.31.2.1 typedef std::shared_ptr<ImageReader> PhotoFinish::ImageReader::ptr

Shared pointer for an ImageReader.

Definition at line 109 of file ImageFile.hh.

6.31.3 Constructor & Destructor Documentation

6.31.3.1 PhotoFinish::ImageReader::ImageReader (const fs::path *fp*) [protected]

Private constructor.

Definition at line 114 of file ImageFile.cc.

6.31.4 Member Function Documentation

6.31.4.1 void PhotoFinish::ImageReader::extract_tags (Image::ptr *img*) [protected]

Extract tags from file.

Definition at line 119 of file ImageFile.cc.

6.31.4.2 ImageReader::ptr PhotoFinish::ImageReader::open (const ImageFilepath & *ifp*) throw UnknownFileType) [static]

Named constructor.

Use the extension of the file path to decide what class to use

Parameters

filepath	File path

Definition at line 137 of file ImageFile.cc.

6.31.4.3 Image::ptr PhotoFinish::ImageReader::read (void) [virtual]

Read the file into an image.

Returns

A new Image object

Definition at line 170 of file ImageFile.cc.

6.31.4.4 virtual Image::ptr PhotoFinish::ImageReader::read (Destination::ptr *dest*) [pure virtual]

Read the file into an image.

Parameters

dest	A Destination object where some information from the file will be placed

Returns

A new Image object

6.31.5 Member Data Documentation

6.31.5.1 const fs::path PhotoFinish::ImageReader::_filepath [protected]

Definition at line 98 of file ImageFile.hh.

6.31.5.2 bool PhotoFinish::ImageReader::_is_open [protected]

Definition at line 99 of file ImageFile.hh.

The documentation for this class was generated from the following files:

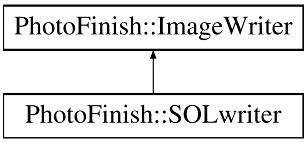
- ImageFile.hh
- ImageFile.cc

6.32 PhotoFinish::ImageWriter Class Reference

Abstract base class for writing image files.

`#include <ImageFile.hh>`

Inheritance diagram for PhotoFinish::ImageWriter:



Public Types

- typedef std::shared_ptr
  < ImageWriter > ptr

  Shared pointer for an ImageWriter.

Public Member Functions

- virtual CMS::Format preferred_format (CMS::Format format)=0

  Modify an LCMS2 pixel format into a "type" that the file format can write.
- virtual void write (Image::ptr img, Destination::ptr dest, bool can_free=false)=0

  Write an image to the file.

Static Public Member Functions

- static ImageWriter::ptr open (const ImageFilepath &ifp) throw (UnknownFileType)

  Named constructor.
- static void add_variables (Destination::ptr dest, multihash &vars)

  Add variables to one of the configuration objects based on destination format.

Protected Member Functions

- ImageWriter (const fs::path fp)

  Private constructor.
- void embed_tags (Image::ptr img) const

Protected Attributes

- const fs::path _filepath
- bool _is_open

6.32.1 Detailed Description

Abstract base class for writing image files.

Definition at line 135 of file ImageFile.hh.

6.32.2 Member Typedef Documentation

6.32.2.1 typedef std::shared_ptr<ImageWriter> PhotoFinish::ImageWriter::ptr

Shared pointer for an ImageWriter.

Definition at line 147 of file ImageFile.hh.

6.32.3 Constructor & Destructor Documentation

6.32.3.1 PhotoFinish::ImageWriter::ImageWriter (const fs::path *fp*) [protected]

Private constructor.

Definition at line 176 of file ImageFile.cc.

6.32.4 Member Function Documentation

6.32.4.1 void PhotoFinish::ImageWriter::add_variables (Destination::ptr *dest*, multihash & *vars*) [static]

Add variables to one of the configuration objects based on destination format.

Definition at line 232 of file ImageFile.cc.

6.32.4.2 void PhotoFinish::ImageWriter::embed_tags (Image::ptr *img*) const [protected]

Definition at line 181 of file ImageFile.cc.

6.32.4.3 ImageWriter::ptr PhotoFinish::ImageWriter::open (const ImageFilepath & *ifp*) throw UnknownFileType) [static]

Named constructor.

Use the extension of the file path to decide what class to use

Parameters

filepath	File path

Definition at line 194 of file ImageFile.cc.

6.32.4.4 virtual CMS::Format PhotoFinish::ImageWriter::preferred_format (CMS::Format *format*) [pure virtual]

Modify an LCMS2 pixel format into a "type" that the file format can write.

Implemented in PhotoFinish::SOLwriter.

6.32.4.5 virtual void PhotoFinish::ImageWriter::write (Image::ptr *img*, Destination::ptr *dest*, bool *can_free* = false) [pure virtual]

Write an image to the file.

Parameters

img	The Image object to write
dest	A Destination object, used for the JPEG/PNG/etc parameters
can_free	Can each row of the image be freed after it is written?

Implemented in PhotoFinish::SOLwriter.

6.32.5 Member Data Documentation

6.32.5.1 const fs::path PhotoFinish::ImageWriter::_filepath [protected]

Definition at line 137 of file ImageFile.hh.

6.32.5.2 bool PhotoFinish::ImageWriter::_is_open [protected]

Definition at line 138 of file ImageFile.hh.

The documentation for this class was generated from the following files:

- ImageFile.hh
- ImageFile.cc

6.33 PhotoFinish::jpeg_destination_state_t Struct Reference

Structure holding information for the ostream writer.

Public Attributes

- JOCTET ∗ buffer
- std::ostream ∗ os
- size_t buffer_size

6.33.1 Detailed Description

Structure holding information for the ostream writer.

Definition at line 106 of file JPEG_iostream.cc.

6.33.2 Member Data Documentation

6.33.2.1 JOCTET∗ PhotoFinish::jpeg_destination_state_t::buffer

Definition at line 107 of file JPEG_iostream.cc.

6.33.2.2 size_t PhotoFinish::jpeg_destination_state_t::buffer_size

Definition at line 109 of file JPEG_iostream.cc.

6.33.2.3 std::ostream∗ PhotoFinish::jpeg_destination_state_t::os

Definition at line 108 of file JPEG_iostream.cc.

The documentation for this struct was generated from the following file:

- JPEG_iostream.cc

6.34 PhotoFinish::jpeg_source_state_t Struct Reference

Structure holding information for the istream reader.

Public Attributes

- JOCTET ∗ buffer
- std::istream ∗ is
- size_t buffer_size

6.34.1 Detailed Description

Structure holding information for the istream reader.

Definition at line 27 of file JPEG_iostream.cc.

6.34.2 Member Data Documentation

6.34.2.1 JOCTET∗ PhotoFinish::jpeg_source_state_t::buffer

Definition at line 28 of file JPEG_iostream.cc.

6.34.2.2 size_t PhotoFinish::jpeg_source_state_t::buffer_size

Definition at line 30 of file JPEG_iostream.cc.

6.34.2.3 std::istream∗ PhotoFinish::jpeg_source_state_t::is

Definition at line 29 of file JPEG_iostream.cc.

The documentation for this struct was generated from the following file:

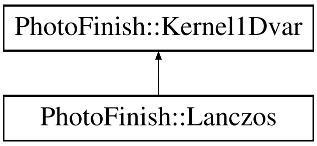
- JPEG_iostream.cc

6.35 PhotoFinish::Kernel1Dvar Class Reference

Creates and stores coefficients for cropping and resizing an image.

```
#include <Kernel1Dvar.hh>
```

Inheritance diagram for PhotoFinish::Kernel1Dvar:



Public Types

- typedef std::shared_ptr
  < Kernel1Dvar > ptr

  Shared pointer for a Kernel1Dvar.

Public Member Functions

- Kernel1Dvar ()

  Emoty constructor.
- ∼Kernel1Dvar ()

  Destructor.
- Image::ptr convolve_h (Image::ptr img, bool can_free=false)

  Convolve an image horizontally with this kernel.

- Image::ptr convolve_v (Image::ptr img, bool can_free=false)

  Convolve an image vertically with this kernel.

Static Public Member Functions

- static ptr create (const D_resize &dr, double from_start, double from_size, unsigned int from_max, double to_size) throw (DestinationError)

  Named constructor.

Protected Member Functions

- Kernel1Dvar (double to_size)

  Private constructor.
- void build (double from_start, double from_size, unsigned int from_max) throw (DestinationError)

  Build the kernel; used by derived classes.
- virtual double range (void) const =0

  The size of this filter.
- virtual SAMPLE eval (double x) const =0 throw (Uninitialised)

  Evaluate the filter at a given point.
- template<typename T , int channels>
  void convolve_h_type_channels (Image::ptr src, Image::ptr dest, bool can_free=false)
- template<typename T >
  void convolve_h_type (Image::ptr src, Image::ptr dest, bool can_free=false)
- template<typename T , int channels>
  void convolve_v_type_channels (Image::ptr src, Image::ptr dest, bool can_free=false)
- template<typename T >
  void convolve_v_type (Image::ptr src, Image::ptr dest, bool can_free=false)

Protected Attributes

- unsigned int ∗ _size
- unsigned int ∗ _start
- SAMPLE ∗∗ _weights
- double _scale
- double _to_size
- unsigned int _to_size_i

6.35.1 Detailed Description

Creates and stores coefficients for cropping and resizing an image.

Definition at line 31 of file Kernel1Dvar.hh.

6.35.2 Member Typedef Documentation

6.35.2.1 typedef std::shared_ptr<Kernel1Dvar> PhotoFinish::Kernel1Dvar::ptr

Shared pointer for a Kernel1Dvar.

Definition at line 64 of file Kernel1Dvar.hh.

6.35.3 Constructor & Destructor Documentation

6.35.3.1 PhotoFinish::Kernel1Dvar::Kernel1Dvar (double *to_size*) [protected]

Private constructor.

Definition at line 37 of file Kernel1Dvar.cc.

6.35.3.2 PhotoFinish::Kernel1Dvar::Kernel1Dvar ( )

Emoty constructor.

Definition at line 32 of file Kernel1Dvar.cc.

6.35.3.3 PhotoFinish::Kernel1Dvar::∼Kernel1Dvar ( )

Destructor.

Definition at line 105 of file Kernel1Dvar.cc.

6.35.4 Member Function Documentation

6.35.4.1 void PhotoFinish::Kernel1Dvar::build (double *from_start,* double *from_size,* unsigned int *from_max*) throw DestinationError) [protected]

Build the kernel; used by derived classes.

Definition at line 48 of file Kernel1Dvar.cc.

6.35.4.2 Image::ptr PhotoFinish::Kernel1Dvar::convolve_h (Image::ptr *img,* bool *can_free* = false)

Convolve an image horizontally with this kernel.

Convolve an image horizontally.

Parameters

img	Source image
can_free	Can each row of the image be freed after it is convolved?

Returns

New image

Definition at line 235 of file Kernel1Dvar.cc.

6.35.4.3 template<typename T > void PhotoFinish::Kernel1Dvar::convolve_h_type (Image::ptr *src,* Image::ptr *dest,* bool *can_free* = false) [protected]

Definition at line 166 of file Kernel1Dvar.cc.

6.35.4.4 template<typename T , int channels> void PhotoFinish::Kernel1Dvar::convolve_h_type_channels (Image::ptr *src,* Image::ptr *dest,* bool *can_free* = false) [protected]

Definition at line 126 of file Kernel1Dvar.cc.

6.35.4.5 Image::ptr PhotoFinish::Kernel1Dvar::convolve_v (Image::ptr *img*, bool *can_free* = `false`)

Convolve an image vertically with this kernel.

Convolve an image vertically.

Parameters

img	Source image
can_free	Can each row of the image be freed after it is convolved?

Returns

New image

Definition at line 405 of file Kernel1Dvar.cc.

6.35.4.6 template<typename T > void PhotoFinish::Kernel1Dvar::convolve_v_type (Image::ptr *src*, Image::ptr *dest*, bool *can_free* = `false`) `[protected]`

Definition at line 336 of file Kernel1Dvar.cc.

6.35.4.7 template<typename T , int channels> void PhotoFinish::Kernel1Dvar::convolve_v_type_channels (Image::ptr *src*, Image::ptr *dest*, bool *can_free* = `false`) `[protected]`

Definition at line 271 of file Kernel1Dvar.cc.

6.35.4.8 Kernel1Dvar::ptr PhotoFinish::Kernel1Dvar::create (const D_resize & *dr*, double *from_start*, double *from_size*, unsigned int *from_max*, double *to_size*) throw DestinationError) `[static]`

Named constructor.

Create a Kernel1Dvar object using the filter name in the D_resize object.

Parameters

dr	A D_resize object which will supply our parameters.
from_start	The starting point of the crop/resample
from_size	The size of the crop/resample
from_max	The size (maximum dimenstion) of the input
to_size	The size of the output

Definition at line 89 of file Kernel1Dvar.cc.

6.35.4.9 virtual SAMPLE PhotoFinish::Kernel1Dvar::eval (double *x*) const throw Uninitialised) `[protected]`, `[pure virtual]`

Evaluate the filter at a given point.

6.35.4.10 virtual double PhotoFinish::Kernel1Dvar::range (void) const `[protected]`, `[pure virtual]`

The size of this filter.

6.35.5 Member Data Documentation

6.35.5.1 double PhotoFinish::Kernel1Dvar::_scale [protected]

Definition at line 35 of file Kernel1Dvar.hh.

6.35.5.2 unsigned int* PhotoFinish::Kernel1Dvar::_size [protected]

Definition at line 33 of file Kernel1Dvar.hh.

6.35.5.3 unsigned int * PhotoFinish::Kernel1Dvar::_start [protected]

Definition at line 33 of file Kernel1Dvar.hh.

6.35.5.4 double PhotoFinish::Kernel1Dvar::_to_size [protected]

Definition at line 35 of file Kernel1Dvar.hh.

6.35.5.5 unsigned int PhotoFinish::Kernel1Dvar::_to_size_i [protected]

Definition at line 36 of file Kernel1Dvar.hh.

6.35.5.6 SAMPLE** PhotoFinish::Kernel1Dvar::_weights [protected]

Definition at line 34 of file Kernel1Dvar.hh.

The documentation for this class was generated from the following files:

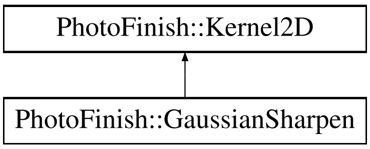
- Kernel1Dvar.hh
- Kernel1Dvar.cc

6.36 PhotoFinish::Kernel2D Class Reference

Creates and stores coefficients for convolving an image.

`#include <Kernel2D.hh>`

Inheritance diagram for PhotoFinish::Kernel2D:



Public Types

- typedef std::shared_ptr< Kernel2D > ptr

    Shared pointer for a Kernel2D.

Public Member Functions

- Kernel2D ()

  Empty constructor.
- ∼Kernel2D ()

  Destructor.
- Image::ptr convolve (Image::ptr img, bool can_free=false)

  Convolve and image with this kernel and produce a new image.

Static Public Member Functions

- static ptr create (const D_sharpen &ds) throw (DestinationError)

  Named constructor.

Protected Member Functions

- Kernel2D (short unsigned int w, short unsigned int h, short unsigned int cx, short unsigned int cy)

  Private constructor for derived classes.
- Kernel2D (short unsigned int size, short unsigned int centre)

  Private constructor for square filters.
- template<typename T >
  void convolve_type (Image::ptr src, Image::ptr dest, bool can_free=false)
- template<typename T , int channels>
  void convolve_type_channels (Image::ptr src, Image::ptr dest, bool can_free=false)

Protected Attributes

- short unsigned int _width
- short unsigned int _height
- short unsigned int _centrex
- short unsigned int _centrey
- SAMPLE ∗∗ _values

6.36.1 Detailed Description

Creates and stores coefficients for convolving an image.

Definition at line 33 of file Kernel2D.hh.

6.36.2 Member Typedef Documentation

6.36.2.1 typedef std::shared_ptr<Kernel2D> PhotoFinish::Kernel2D::ptr

Shared pointer for a Kernel2D.

Definition at line 52 of file Kernel2D.hh.

6.36.3 Constructor & Destructor Documentation

6.36.3.1 PhotoFinish::Kernel2D::Kernel2D (short unsigned int *w*, short unsigned int *h*, short unsigned int *cx*, short unsigned int *cy*) `[protected]`

Private constructor for derived classes.

Definition at line 34 of file Kernel2D.cc.

6.36.3.2 PhotoFinish::Kernel2D::Kernel2D (short unsigned int *size*, short unsigned int *centre*) `[protected]`

Private constructor for square filters.

Definition at line 44 of file Kernel2D.cc.

6.36.3.3 PhotoFinish::Kernel2D::Kernel2D ( )

Empty constructor.

Definition at line 28 of file Kernel2D.cc.

6.36.3.4 PhotoFinish::Kernel2D::∼Kernel2D ( )

Destructor.

Definition at line 58 of file Kernel2D.cc.

6.36.4 Member Function Documentation

6.36.4.1 Image::ptr PhotoFinish::Kernel2D::convolve (Image::ptr *img*, bool *can_free* = `false`)

Convolve and image with this kernel and produce a new image.

Parameters

img	Source image
can_free	Can each row of the image be freed after it is convolved?

Returns

New image

Definition at line 200 of file Kernel2D.cc.

6.36.4.2 template<typename T > void PhotoFinish::Kernel2D::convolve_type (Image::ptr *src*, Image::ptr *dest*, bool *can_free* = `false`) `[protected]`

Definition at line 132 of file Kernel2D.cc.

6.36.4.3 template<typename T , int channels> void PhotoFinish::Kernel2D::convolve_type_channels (Image::ptr *src*, Image::ptr *dest*, bool *can_free* = `false`) `[protected]`

Definition at line 68 of file Kernel2D.cc.

6.36.4.4 Kernel2D::ptr PhotoFinish::Kernel2D::create (const D_sharpen & *ds*) throw DestinationError) [static]

Named constructor.

Create a Kernel2D object using the parameters in the D_sharpen object.

Parameters

ds	A D_sharpen object which will supply our parameters.

Definition at line 54 of file Kernel2D.cc.

6.36.5 Member Data Documentation

6.36.5.1 short unsigned int PhotoFinish::Kernel2D::_centrex [protected]

Definition at line 35 of file Kernel2D.hh.

6.36.5.2 short unsigned int PhotoFinish::Kernel2D::_centrey [protected]

Definition at line 35 of file Kernel2D.hh.

6.36.5.3 short unsigned int PhotoFinish::Kernel2D::_height [protected]

Definition at line 35 of file Kernel2D.hh.

6.36.5.4 SAMPLE** PhotoFinish::Kernel2D::_values [protected]

Definition at line 36 of file Kernel2D.hh.

6.36.5.5 short unsigned int PhotoFinish::Kernel2D::_width [protected]

Definition at line 35 of file Kernel2D.hh.

The documentation for this class was generated from the following files:

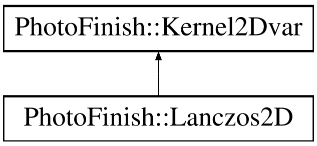
- Kernel2D.hh
- Kernel2D.cc

6.37 PhotoFinish::Kernel2Dvar Class Reference

Creates and stores coefficients for cropping and resizing an image.

```
#include <Kernel2Dvar.hh>
```

Inheritance diagram for PhotoFinish::Kernel2Dvar:



Public Types

- typedef std::shared_ptr
  < Kernel2Dvar > ptr

  Shared pointer for a Kernel2Dvar.

Public Member Functions

- Kernel2Dvar ()

  Emoty constructor.
- ∼Kernel2Dvar ()

  Destructor.
- Image::ptr convolve (Image::ptr img, bool can_free=false)

  Convolve an image with this kernel.

Static Public Member Functions

- static ptr create (const D_resize &dr, double from_startx, double from_starty, double from_sizex, double from_sizey, unsigned int from_width, unsigned int from_height, double to_width, double to_height) throw (DestinationError)

  Named constructor.

Protected Member Functions

- Kernel2Dvar (double from_startx, double from_starty, double from_sizex, double from_sizey, unsigned int from_width, unsigned int from_height, double to_width, double to_height)

  Private constructor.
- virtual double radius (void) const =0

  The size of this filter.
- virtual SAMPLE eval (double r) const =0 throw (Uninitialised)

  Evaluate the filter at a given point.
- template<typename T , int channels>
  void convolve_type_channels (Image::ptr src, Image::ptr dest, bool can_free=false)
- template<typename T >
  void convolve_type (Image::ptr src, Image::ptr dest, bool can_free=false)

Protected Attributes

- double _startx
- double _starty
- double _scalex
- double _scaley
- unsigned int _from_width
- unsigned int _from_height
- double _to_width
- double _to_height
- unsigned int _to_width_i
- unsigned int _to_height_i

6.37.1 Detailed Description

Creates and stores coefficients for cropping and resizing an image.

Definition at line 31 of file Kernel2Dvar.hh.

6.37.2 Member Typedef Documentation

6.37.2.1 typedef std::shared_ptr<Kernel2Dvar> PhotoFinish::Kernel2Dvar::ptr

Shared pointer for a Kernel2Dvar.

Definition at line 59 of file Kernel2Dvar.hh.

6.37.3 Constructor & Destructor Documentation

6.37.3.1 PhotoFinish::Kernel2Dvar::Kernel2Dvar (double *from_startx,* double *from_starty,* double *from_sizex,* double *from_sizey,* unsigned int *from_width,* unsigned int *from_height,* double *to_width,* double *to_height*) [protected]

Private constructor.

Definition at line 36 of file Kernel2Dvar.cc.

6.37.3.2 PhotoFinish::Kernel2Dvar::Kernel2Dvar ( )

Emoty constructor.

Definition at line 33 of file Kernel2Dvar.cc.

6.37.3.3 PhotoFinish::Kernel2Dvar::∼Kernel2Dvar ( )

Destructor.

Definition at line 70 of file Kernel2Dvar.cc.

6.37.4 Member Function Documentation

6.37.4.1 Image::ptr PhotoFinish::Kernel2Dvar::convolve (Image::ptr *img,* bool *can_free* = false)

Convolve an image with this kernel.

Convolve an image.

Parameters

img	Source image
can_free	Can each row of the image be freed after it is convolved?

Returns

New image

Definition at line 224 of file Kernel2Dvar.cc.

6.37.4.2 template<typename T > void PhotoFinish::Kernel2Dvar::convolve_type (Image::ptr *src*, Image::ptr *dest*, bool *can_free* = false) [protected]

Definition at line 187 of file Kernel2Dvar.cc.

6.37.4.3 template<typename T , int channels> void PhotoFinish::Kernel2Dvar::convolve_type_channels (Image::ptr *src*, Image::ptr *dest*, bool *can_free* = false) [protected]

Definition at line 75 of file Kernel2Dvar.cc.

6.37.4.4 Kernel2Dvar::ptr PhotoFinish::Kernel2Dvar::create (const D_resize & *dr*, double *from_startx*, double *from_starty*, double *from_sizex*, double *from_sizey*, unsigned int *from_width*, unsigned int *from_height*, double *to_width*, double *to_height*) throw DestinationError) [static]

Named constructor.

Create a Kernel2Dvar object using the filter name in the D_resize object.

Parameters

dr	A D_resize object which will supply our parameters.
from_start	The starting point of the crop/resample
from_size	The size of the crop/resample
from_max	The size (maximum dimension) of the input
to_size	The size of the output

Definition at line 50 of file Kernel2Dvar.cc.

6.37.4.5 virtual SAMPLE PhotoFinish::Kernel2Dvar::eval (double *r*) const throw Uninitialised) [protected], [pure virtual]

Evaluate the filter at a given point.

6.37.4.6 virtual double PhotoFinish::Kernel2Dvar::radius (void) const [protected], [pure virtual]

The size of this filter.

6.37.5 Member Data Documentation

6.37.5.1 unsigned int PhotoFinish::Kernel2Dvar::_from_height [protected]

Definition at line 35 of file Kernel2Dvar.hh.

6.37.5.2 unsigned int PhotoFinish::Kernel2Dvar::_from_width [protected]

Definition at line 35 of file Kernel2Dvar.hh.

6.37.5.3 double PhotoFinish::Kernel2Dvar::_scalex [protected]

Definition at line 34 of file Kernel2Dvar.hh.

6.37.5.4 double PhotoFinish::Kernel2Dvar::_scaley [protected]

Definition at line 34 of file Kernel2Dvar.hh.

6.37.5.5 double PhotoFinish::Kernel2Dvar::_startx [protected]

Definition at line 33 of file Kernel2Dvar.hh.

6.37.5.6 double PhotoFinish::Kernel2Dvar::_starty [protected]

Definition at line 33 of file Kernel2Dvar.hh.

6.37.5.7 double PhotoFinish::Kernel2Dvar::_to_height [protected]

Definition at line 36 of file Kernel2Dvar.hh.

6.37.5.8 unsigned int PhotoFinish::Kernel2Dvar::_to_height_i [protected]

Definition at line 37 of file Kernel2Dvar.hh.

6.37.5.9 double PhotoFinish::Kernel2Dvar::_to_width [protected]

Definition at line 36 of file Kernel2Dvar.hh.

6.37.5.10 unsigned int PhotoFinish::Kernel2Dvar::_to_width_i [protected]

Definition at line 37 of file Kernel2Dvar.hh.

The documentation for this class was generated from the following files:

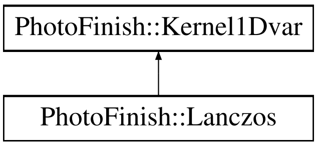
- Kernel2Dvar.hh
- Kernel2Dvar.cc

6.38 PhotoFinish::Lanczos Class Reference

Lanczos filter.

```
#include <Kernel1Dvar.hh>
```

Inheritance diagram for PhotoFinish::Lanczos:



Public Member Functions

- Lanczos ()

  Empty constructor.
- Lanczos (const D_resize &dr, double from_start, double from_size, unsigned int from_max, double to_size)

  Constructor.

Additional Inherited Members

6.38.1 Detailed Description

Lanczos filter.

Definition at line 101 of file Kernel1Dvar.hh.

6.38.2 Constructor & Destructor Documentation

6.38.2.1 PhotoFinish::Lanczos::Lanczos ( )

Empty constructor.

Definition at line 441 of file Kernel1Dvar.cc.

6.38.2.2 PhotoFinish::Lanczos::Lanczos (const D_resize & *dr*, double *from_start*, double *from_size*, unsigned int *from_max*, double *to_size*)

Constructor.

Parameters

dr	A D_resize object which will supply our parameters.
horiz	Will the kernel run in horizontal (true) or vertical direction?
from_start	The starting point of the crop/resample
from_size	The size of the crop/resample
from_max	The size (maximum dimenstion) of the input
to_size	The size of the output

Definition at line 445 of file Kernel1Dvar.cc.

The documentation for this class was generated from the following files:

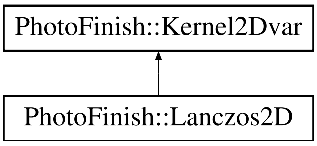
- Kernel1Dvar.hh
- Kernel1Dvar.cc

6.39 PhotoFinish::Lanczos2D Class Reference

Lanczos filter.

```
#include <Kernel2Dvar.hh>
```

Inheritance diagram for PhotoFinish::Lanczos2D:



Public Member Functions

- Lanczos2D ()

  Empty constructor.

- Lanczos2D (const D_resize &dr, double from_startx, double from_starty, double from_sizex, double from_sizey, unsigned int from_width, unsigned int from_height, double to_width, double to_height)

  Constructor.

Additional Inherited Members

6.39.1 Detailed Description

Lanczos filter.

Definition at line 92 of file Kernel2Dvar.hh.

6.39.2 Constructor & Destructor Documentation

6.39.2.1 PhotoFinish::Lanczos2D::Lanczos2D ( )

Empty constructor.

Definition at line 260 of file Kernel2Dvar.cc.

6.39.2.2 PhotoFinish::Lanczos2D::Lanczos2D (const D_resize & *dr,* double *from_startx,* double *from_starty,* double *from_sizex,* double *from_sizey,* unsigned int *from_width,* unsigned int *from_height,* double *to_width,* double *to_height*)

Constructor.

Parameters

dr	A D_resize object which will supply our parameters.
horiz	Will the kernel run in horizontal (true) or vertical direction?
from_start	The starting point of the crop/resample
from_size	The size of the crop/resample
from_max	The size (maximum dimenstion) of the input
to_size	The size of the output

Definition at line 264 of file Kernel2Dvar.cc.

The documentation for this class was generated from the following files:

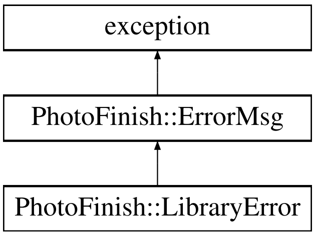
- Kernel2Dvar.hh
- Kernel2Dvar.cc

6.40 PhotoFinish::LibraryError Class Reference

Library exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::LibraryError:



Public Member Functions

- LibraryError (const std::string &l, const std::string &m)

  Constructor.
- virtual const char ∗ what () const throw ()

Additional Inherited Members

6.40.1 Detailed Description

Library exception.

Definition at line 283 of file Exception.hh.

6.40.2 Constructor & Destructor Documentation

6.40.2.1 PhotoFinish::LibraryError::LibraryError (const std::string & *l*, const std::string & *m*) `[inline]`

Constructor.

Parameters

l	Library name
m	Error message

Definition at line 293 of file Exception.hh.

6.40.3 Member Function Documentation

6.40.3.1 virtual const char∗ PhotoFinish::LibraryError::what ( ) const throw) `[inline]`, `[virtual]`

Implements PhotoFinish::ErrorMsg.

Definition at line 297 of file Exception.hh.

The documentation for this class was generated from the following file:

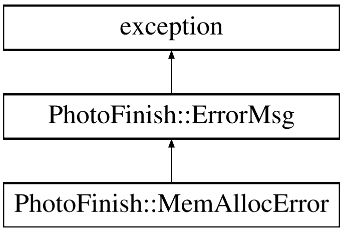
- Exception.hh

6.41 PhotoFinish::MemAllocError Class Reference

Memory allocation exception.

`#include <Exception.hh>`

Inheritance diagram for PhotoFinish::MemAllocError:



Public Member Functions

- MemAllocError (const std::string &m)

  Constructor.
- const char ∗ what () const throw ()

Additional Inherited Members

6.41.1 Detailed Description

Memory allocation exception.

Definition at line 134 of file Exception.hh.

6.41.2 Constructor & Destructor Documentation

6.41.2.1 PhotoFinish::MemAllocError::MemAllocError (const std::string & *m*) `[inline]`

Constructor.

Parameters

m	Error message

Definition at line 140 of file Exception.hh.

6.41.3 Member Function Documentation

6.41.3.1 const char∗ PhotoFinish::MemAllocError::what ( ) const throw) `[inline], [virtual]`

Implements PhotoFinish::ErrorMsg.

Definition at line 144 of file Exception.hh.

The documentation for this class was generated from the following file:

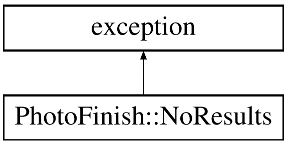
- Exception.hh

6.42 PhotoFinish::NoResults Class Reference

No results exception.

`#include <Exception.hh>`

Inheritance diagram for PhotoFinish::NoResults:



Public Member Functions

- NoResults (const std::string &c, const std::string &m)

Constructor.

- virtual const char ∗ what () const throw ()

Protected Attributes

- const std::string _class
- const std::string _method

6.42.1 Detailed Description

No results exception.

Definition at line 78 of file Exception.hh.

6.42.2 Constructor & Destructor Documentation

6.42.2.1 PhotoFinish::NoResults::NoResults (const std::string & *c*, const std::string & *m*) [inline]

Constructor.

Parameters

c	Class name
m	Method name

Definition at line 88 of file Exception.hh.

6.42.3 Member Function Documentation

6.42.3.1 virtual const char∗ PhotoFinish::NoResults::what ( ) const throw) [inline], [virtual]

Definition at line 92 of file Exception.hh.

6.42.4 Member Data Documentation

6.42.4.1 const std::string PhotoFinish::NoResults::_class [protected]

Definition at line 80 of file Exception.hh.

6.42.4.2 const std::string PhotoFinish::NoResults::_method [protected]

Definition at line 80 of file Exception.hh.

The documentation for this class was generated from the following file:

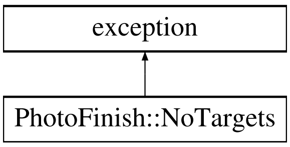
- Exception.hh

6.43 PhotoFinish::NoTargets Class Reference

No targets exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::NoTargets:



Public Member Functions

- NoTargets (const std::string &d)

    Constructor.
- virtual const char ∗ what () const throw ()

Protected Attributes

- const std::string _destination

6.43.1 Detailed Description

No targets exception.

Definition at line 98 of file Exception.hh.

6.43.2 Constructor & Destructor Documentation

6.43.2.1 PhotoFinish::NoTargets::NoTargets (const std::string & *d*) [inline]

Constructor.

Parameters

d	Name of destination that has no targets

Definition at line 107 of file Exception.hh.

6.43.3 Member Function Documentation

6.43.3.1 virtual const char∗ PhotoFinish::NoTargets::what ( ) const throw) [inline], [virtual]

Definition at line 111 of file Exception.hh.

6.43.4 Member Data Documentation

6.43.4.1 const std::string PhotoFinish::NoTargets::_destination [protected]

Definition at line 100 of file Exception.hh.

The documentation for this class was generated from the following file:

- Exception.hh

6.44 PhotoFinish::PNGreader_cb Struct Reference

```
#include <PNGreader_cb.hh>
```

Public Member Functions

- PNGreader_cb (Destination::ptr d)
- void info (png_structp png, png_infop info)
- void row (png_structp png, png_bytep row_data, png_uint_32 row_num, int pass)
- void end (png_structp png, png_infop info)

Public Attributes

- Destination::ptr _destination
- Image::ptr _image

6.44.1 Detailed Description

Definition at line 28 of file PNGreader_cb.hh.

6.44.2 Constructor & Destructor Documentation

6.44.2.1 PhotoFinish::PNGreader_cb::PNGreader_cb (Destination::ptr *d*)

Definition at line 23 of file PNGreader_cb.cc.

6.44.3 Member Function Documentation

6.44.3.1 void PhotoFinish::PNGreader_cb::end (png_structp *png,* png_infop *info*)

Definition at line 131 of file PNGreader_cb.cc.

6.44.3.2 void PhotoFinish::PNGreader_cb::info (png_structp *png,* png_infop *info*)

Definition at line 27 of file PNGreader_cb.cc.

6.44.3.3 void PhotoFinish::PNGreader_cb::row (png_structp *png,* png_bytep *row_data,* png_uint_32 *row_num,* int *pass*)

Definition at line 120 of file PNGreader_cb.cc.

6.44.4 Member Data Documentation

6.44.4.1 Destination::ptr PhotoFinish::PNGreader_cb::_destination

Definition at line 29 of file PNGreader_cb.hh.

6.44.4.2 Image::ptr PhotoFinish::PNGreader_cb::_image

Definition at line 30 of file PNGreader_cb.hh.

The documentation for this struct was generated from the following files:

- PNGreader_cb.hh
- PNGreader_cb.cc

6.45 CMS::Profile Class Reference

Wrap LCMS2's cmsHPROFILE.

`#include <CMS.hh>`

Public Types

- typedef std::shared_ptr< Profile > ptr

  Shared pointer typedef.

Public Member Functions

- Profile ()

  Empty constructor.
- Profile (const Profile &other)

  Copy constructor.
- Profile (fs::path filepath)

  Constructor from file path.
- Profile (const void ∗data, cmsUInt32Number size)

  Constructor from memory.
- Profile (std::istream stream)

  Constructor from an istream.
- ∼Profile ()

  Deconstructor.
- operator cmsHPROFILE () const

  Cast to a profile handle for direct use with LCMS2.
- void write_tag (cmsTagSignature sig, std::string lang, std::string cc, std::string text)
- void write_tag (cmsTagSignature sig, std::string lang, std::string cc, std::wstring text)
- std::string read_info (cmsInfoType type, std::string lang, std::string cc) const
- std::wstring read_info_wide (cmsInfoType type, std::string lang, std::string cc) const
- void save_to_mem (void ∗&dest, unsigned int &size) const

Static Public Member Functions

- static ptr Lab4 (void)

  Named constructor.
- static ptr sRGB (void)

  Named constructor.
- static ptr sGrey (void)

  Named constructor.

Friends

- class __gnu_cxx::new_allocator< Profile >

6.45.1 Detailed Description

Wrap LCMS2's cmsHPROFILE.

Definition at line 37 of file CMS.hh.

6.45.2 Member Typedef Documentation

6.45.2.1 typedef std::shared_ptr<Profile> CMS::Profile::ptr

Shared pointer typedef.

Definition at line 71 of file CMS.hh.

6.45.3 Constructor & Destructor Documentation

6.45.3.1 CMS::Profile::Profile ( )

Empty constructor.

Definition at line 34 of file CMS.cc.

6.45.3.2 CMS::Profile::Profile (const Profile & *other*)

Copy constructor.

Definition at line 39 of file CMS.cc.

6.45.3.3 CMS::Profile::Profile (fs::path *filepath*)

Constructor from file path.

Definition at line 52 of file CMS.cc.

6.45.3.4 CMS::Profile::Profile (const void ∗ *data,* cmsUInt32Number *size*)

Constructor from memory.

Definition at line 57 of file CMS.cc.

6.45.3.5 CMS::Profile::Profile (std::istream *stream*)

Constructor from an istream.

Definition at line 62 of file CMS.cc.

6.45.3.6 CMS::Profile::∼Profile ( )

Deconstructor.

Definition at line 67 of file CMS.cc.

6.45.4 Member Function Documentation

6.45.4.1 Profile::ptr CMS::Profile::Lab4 (void) [static]

Named constructor.

Definition at line 72 of file CMS.cc.

6.45.4.2 CMS::Profile::operator cmsHPROFILE ( ) const [inline]

Cast to a profile handle for direct use with LCMS2.

Definition at line 68 of file CMS.hh.

6.45.4.3 std::string CMS::Profile::read_info (cmsInfoType *type*, std::string *lang*, std::string *cc*) const

Definition at line 118 of file CMS.cc.

6.45.4.4 std::wstring CMS::Profile::read_info_wide (cmsInfoType *type*, std::string *lang*, std::string *cc*) const

Definition at line 131 of file CMS.cc.

6.45.4.5 void CMS::Profile::save_to_mem (void ∗& *dest*, unsigned int & *size*) const

Definition at line 144 of file CMS.cc.

6.45.4.6 Profile::ptr CMS::Profile::sGrey (void) [static]

Named constructor.

Definition at line 80 of file CMS.cc.

6.45.4.7 Profile::ptr CMS::Profile::sRGB (void) [static]

Named constructor.

Definition at line 76 of file CMS.cc.

6.45.4.8 void CMS::Profile::write_tag (cmsTagSignature *sig*, std::string *lang*, std::string *cc*, std::string *text*)

Definition at line 100 of file CMS.cc.

6.45.4.9 void CMS::Profile::write_tag (cmsTagSignature *sig*, std::string *lang*, std::string *cc*, std::wstring *text*)

Definition at line 109 of file CMS.cc.

6.45.5 Friends And Related Function Documentation

6.45.5.1 friend class __gnu_cxx::new_allocator< Profile > [friend]

Definition at line 46 of file CMS.hh.

The documentation for this class was generated from the following files:

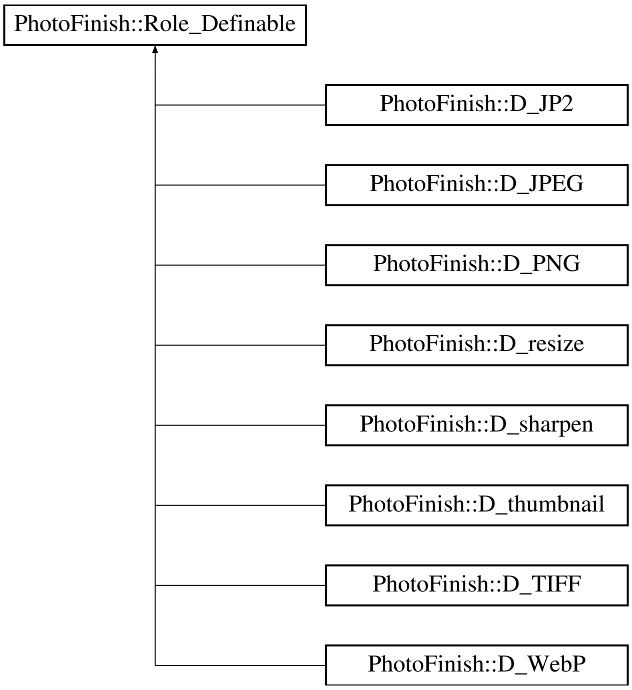
- CMS.hh
- CMS.cc

6.46 PhotoFinish::Role_Definable Class Reference

Base class for adding "definable" attribute.

`#include <Definable.hh>`

Inheritance diagram for PhotoFinish::Role_Definable:



Public Member Functions

- Role_Definable ()

  Empty constructor.
- const bool defined (void) const

  Is this object defined?

Protected Member Functions

- void set_defined (bool v=true)

  Set this object as 'defined' (or not)
- void undefine (void)

  Undefine the object.

Protected Attributes

- bool _defined

Friends

- bool defined (const Role_Definable &obj)

6.46.1 Detailed Description

Base class for adding "definable" attribute.

Definition at line 99 of file Definable.hh.

6.46.2 Constructor & Destructor Documentation

6.46.2.1 PhotoFinish::Role_Definable::Role_Definable ( ) [inline]

Empty constructor.

Sets defined to false

Definition at line 114 of file Definable.hh.

6.46.3 Member Function Documentation

6.46.3.1 const bool PhotoFinish::Role_Definable::defined (void) const [inline]

Is this object defined?

Definition at line 119 of file Definable.hh.

6.46.3.2 void PhotoFinish::Role_Definable::set_defined (bool *v* = true) [inline], [protected]

Set this object as 'defined' (or not)

Definition at line 104 of file Definable.hh.

6.46.3.3 void PhotoFinish::Role_Definable::undefine (void) [inline], [protected]

Undefine the object.

Definition at line 107 of file Definable.hh.

6.46.4 Friends And Related Function Documentation

6.46.4.1 bool defined (const Role_Definable & *obj*) [friend]

Definition at line 121 of file Definable.hh.

6.46.5 Member Data Documentation

6.46.5.1 bool PhotoFinish::Role_Definable::_defined [protected]

Definition at line 101 of file Definable.hh.

The documentation for this class was generated from the following file:

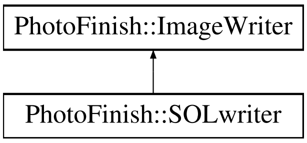
- Definable.hh

6.47 PhotoFinish::SOLwriter Class Reference

Write the boot logo files for use on Motorola Atrix 4G and possibly other phones.

```
#include <ImageFile.hh>
```

Inheritance diagram for PhotoFinish::SOLwriter:



Public Member Functions

- SOLwriter (const fs::path filepath)
- CMS::Format preferred_format (CMS::Format format)

  Modify an LCMS2 pixel format into a "type" that the file format can write.
- void write (Image::ptr img, Destination::ptr dest, bool can_free=false)

  Write an image to the file.

Additional Inherited Members

6.47.1 Detailed Description

Write the boot logo files for use on Motorola Atrix 4G and possibly other phones.

I haven't been able to find any documentation about this format. It starts with the ASCII string "SOL:" followed by eight null bytes. Then comes the width and height as big-endian 32-bit values. The image data is as uncompressed 5-6-5 bit pixels i.e 16 bits per pixel. No footer.

Definition at line 307 of file ImageFile.hh.

6.47.2 Constructor & Destructor Documentation

6.47.2.1 PhotoFinish::SOLwriter::SOLwriter (const fs::path *filepath*)

Definition at line 26 of file SOLwriter.cc.

6.47.3 Member Function Documentation

6.47.3.1 CMS::Format PhotoFinish::SOLwriter::preferred_format (CMS::Format *format*) [virtual]

Modify an LCMS2 pixel format into a "type" that the file format can write.

Implements PhotoFinish::ImageWriter.

Definition at line 30 of file SOLwriter.cc.

6.47.3.2 void PhotoFinish::SOLwriter::write (Image::ptr *img*, Destination::ptr *dest*, bool *can_free* = false) [virtual]

Write an image to the file.

Parameters

img	The Image object to write
dest	A Destination object, used for the JPEG/PNG/etc parameters
can_free	Can each row of the image be freed after it is written?

Implements PhotoFinish::ImageWriter.

Definition at line 53 of file SOLwriter.cc.

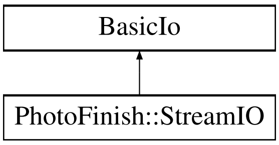
The documentation for this class was generated from the following files:

- ImageFile.hh
- SOLwriter.cc

6.48 PhotoFinish::StreamIO Class Reference

```
#include <StreamIO.hh>
```

Inheritance diagram for PhotoFinish::StreamIO:



Public Member Functions

- StreamIO (const std::iostream &s)

    Constructor.
- int open ()
- int close ()
- long write (const unsigned char ∗data, long wcount)
- long write (Exiv2::BasicIo &src)
- int putb (unsigned char data)
- Exiv2::DataBuf read (long rcount)
- long read (unsigned char ∗buf, long rcount)
- int getb ()
- void transfer (Exiv2::BasicIo &src)
- int seek (uint64_t offset, Position pos)
- int seek (long offset, Position pos)
- unsigned char ∗ mmap (bool isWriteable=false)
- int munmap ()
- long tell () const
- long size () const
- bool isopen () const
- int error () const
- bool eof () const
- std::string path () const
- std::wstring wpath () const
- Exiv2::BasicIo::AutoPtr temporary () const

6.48.1 Detailed Description

Definition at line 33 of file StreamIO.hh.

6.48.2 Constructor & Destructor Documentation

6.48.2.1 PhotoFinish::StreamIO::StreamIO (const std::iostream & *s*)

Constructor.

Definition at line 23 of file StreamIO.cc.

6.48.3 Member Function Documentation

6.48.3.1 int PhotoFinish::StreamIO::close ( )

Definition at line 33 of file StreamIO.cc.

6.48.3.2 bool PhotoFinish::StreamIO::eof ( ) const

Definition at line 125 of file StreamIO.cc.

6.48.3.3 int PhotoFinish::StreamIO::error ( ) const

Definition at line 121 of file StreamIO.cc.

6.48.3.4 int PhotoFinish::StreamIO::getb ( )

Definition at line 81 of file StreamIO.cc.

6.48.3.5 bool PhotoFinish::StreamIO::isopen ( ) const

Definition at line 117 of file StreamIO.cc.

6.48.3.6 unsigned char ∗ PhotoFinish::StreamIO::mmap (bool *isWriteable =* `false`)

Definition at line 101 of file StreamIO.cc.

6.48.3.7 int PhotoFinish::StreamIO::munmap ( )

Definition at line 105 of file StreamIO.cc.

6.48.3.8 int PhotoFinish::StreamIO::open ( )

Definition at line 27 of file StreamIO.cc.

6.48.3.9 std::string PhotoFinish::StreamIO::path ( ) const

Definition at line 129 of file StreamIO.cc.

6.48.3.10 int PhotoFinish::StreamIO::putb (unsigned char *data*)

Definition at line 58 of file StreamIO.cc.

6.48.3.11 Exiv2::DataBuf PhotoFinish::StreamIO::read (long *rcount*)

Definition at line 63 of file StreamIO.cc.

6.48.3.12 long PhotoFinish::StreamIO::read (unsigned char ∗ *buf,* long *rcount*)

Definition at line 73 of file StreamIO.cc.

6.48.3.13 int PhotoFinish::StreamIO::seek (uint64_t *offset,* Position *pos*)

Definition at line 89 of file StreamIO.cc.

6.48.3.14 int PhotoFinish::StreamIO::seek (long *offset,* Position *pos*)

Definition at line 95 of file StreamIO.cc.

6.48.3.15 long PhotoFinish::StreamIO::size (void) const

Definition at line 113 of file StreamIO.cc.

6.48.3.16 long PhotoFinish::StreamIO::tell ( ) const

Definition at line 109 of file StreamIO.cc.

6.48.3.17 Exiv2::BasicIo::AutoPtr PhotoFinish::StreamIO::temporary ( ) const

Definition at line 137 of file StreamIO.cc.

6.48.3.18 void PhotoFinish::StreamIO::transfer (Exiv2::BasicIo & *src*)

Definition at line 86 of file StreamIO.cc.

6.48.3.19 std::wstring PhotoFinish::StreamIO::wpath ( ) const

Definition at line 133 of file StreamIO.cc.

6.48.3.20 long PhotoFinish::StreamIO::write (const unsigned char ∗ *data,* long *wcount*)

Definition at line 37 of file StreamIO.cc.

6.48.3.21 long PhotoFinish::StreamIO::write (Exiv2::BasicIo & *src*)

Definition at line 45 of file StreamIO.cc.

The documentation for this class was generated from the following files:

- StreamIO.hh
- StreamIO.cc

6.49 PhotoFinish::Tags Class Reference

Reads and holds tag information.

`#include <Tags.hh>`

Public Types

- typedef std::shared_ptr< Tags > ptr

  Shared pointer for a Tags object.

Public Member Functions

- Tags ()

  Empty Constructor.
- Tags (const Tags &other)

  Copy constructor.
- Tags (const fs::path &filepath)

  Constructor with a filepath from which to load tags (calls Load)
- ptr dupe (void) const

  Duplicate the tags.
- void add_searchpath (fs::path path)
- multihash & variables (void)

  The map of variables.
- Exiv2::ExifData & EXIFtags (void)

  The Exiv2::ExifData object.
- Exiv2::IptcData & IPTCtags (void)

  The Exiv2::IptcData object.
- Exiv2::XmpData & XMPtags (void)

  The Exiv2::XmpData object.
- bool try_load (fs::path filepath)
- void load (fs::path filepath)

  Load tags from supplied file path.
- void copy_from (Image::ptr img)

  Copy EXIF/IPTC/XMP tags from an image.
- void make_thumbnail (Image::ptr img, const D_thumbnail &dt)

  Create a thumbnail from the supplied image.
- void add_resolution (Image::ptr img)
- void copy_to (Image::ptr img) const

  Copy EXIF/IPTC/XMP tags to an image.

6.49.1 Detailed Description

Reads and holds tag information.

Definition at line 41 of file Tags.hh.

6.49.2 Member Typedef Documentation

6.49.2.1 typedef std::shared_ptr<Tags> PhotoFinish::Tags::ptr

Shared pointer for a Tags object.

Definition at line 60 of file Tags.hh.

6.49.3 Constructor & Destructor Documentation

6.49.3.1 PhotoFinish::Tags::Tags ( )

Empty Constructor.

Definition at line 33 of file Tags.cc.

6.49.3.2 PhotoFinish::Tags::Tags (const Tags & *other*)

Copy constructor.

Definition at line 36 of file Tags.cc.

6.49.3.3 PhotoFinish::Tags::Tags (const fs::path & *filepath*)

Constructor with a filepath from which to load tags (calls Load)

Definition at line 44 of file Tags.cc.

6.49.4 Member Function Documentation

6.49.4.1 void PhotoFinish::Tags::add_resolution (Image::ptr *img*)

Definition at line 219 of file Tags.cc.

6.49.4.2 void PhotoFinish::Tags::add_searchpath (fs::path *path*) [inline]

Definition at line 65 of file Tags.hh.

6.49.4.3 void PhotoFinish::Tags::copy_from (Image::ptr *img*)

Copy EXIF/IPTC/XMP tags from an image.

Definition at line 171 of file Tags.cc.

6.49.4.4 void PhotoFinish::Tags::copy_to (Image::ptr *img*) const

Copy EXIF/IPTC/XMP tags to an image.

Definition at line 242 of file Tags.cc.

6.49.4.5 Tags::ptr PhotoFinish::Tags::dupe (void) const

Duplicate the tags.

Definition at line 48 of file Tags.cc.

6.49.4.6 Exiv2::ExifData& PhotoFinish::Tags::EXIFtags (void) [inline]

The Exiv2::ExifData object.

Definition at line 71 of file Tags.hh.

6.49.4.7 Exiv2::IptcData& PhotoFinish::Tags::IPTCtags (void) [inline]

The Exiv2::IptcData object.

Definition at line 74 of file Tags.hh.

6.49.4.8 void PhotoFinish::Tags::load (fs::path *filepath*)

Load tags from supplied file path.

Definition at line 68 of file Tags.cc.

6.49.4.9 void PhotoFinish::Tags::make_thumbnail (Image::ptr *img,* const D_thumbnail & *dt*)

Create a thumbnail from the supplied image.

Definition at line 182 of file Tags.cc.

6.49.4.10 bool PhotoFinish::Tags::try_load (fs::path *filepath*)

Try to load tags from a file, looking in the search paths

Returns

if the file was found and loaded

Definition at line 57 of file Tags.cc.

6.49.4.11 multihash& PhotoFinish::Tags::variables (void) [inline]

The map of variables.

Definition at line 68 of file Tags.hh.

6.49.4.12 Exiv2::XmpData& PhotoFinish::Tags::XMPtags (void) [inline]

The Exiv2::XmpData object.

Definition at line 77 of file Tags.hh.

The documentation for this class was generated from the following files:

- Tags.hh
- Tags.cc

6.50 CMS::Transform Class Reference

Wrap LCMS2's transform object.

```
#include <CMS.hh>
```

Public Types

- typedef std::shared_ptr
  < Transform > ptr

Public Member Functions

- Transform (Profile::ptr input, const Format &informat, Profile::ptr output, const Format &outformat, Intent intent, cmsUInt32Number flags)

  Construct a transform from two profiles and formats.
- Transform (std::vector< Profile::ptr > profile, const Format &informat, const Format &outformat, Intent intent, cmsUInt32Number flags)

  Construct a transform from multiple profiles.
- ∼Transform ()

  Deconstructor.
- Format input_format (void) const

  Get the input format.
- Format output_format (void) const

  Get the output format.
- void change_formats (const Format &informat, const Format &outformat)

  Change the input and output formats.
- Profile::ptr device_link (double version, cmsUInt32Number flags) const

  Create a device link profile from this transform.
- void transform_buffer (const void ∗input, void ∗output, cmsUInt32Number size) const

Static Public Member Functions

- static ptr Proofing (Profile::ptr input, const Format &informat, Profile::ptr output, const Format &outformat, Profile::ptr proofing, Intent intent, Intent proofing_intent, cmsUInt32Number flags)

  Named constructor for creating a proofing transform.

Friends

- class __gnu_cxx::new_allocator< Transform >

6.50.1 Detailed Description

Wrap LCMS2's transform object.

Definition at line 353 of file CMS.hh.

6.50.2 Member Typedef Documentation

6.50.2.1 typedef std::shared_ptr<Transform> CMS::Transform::ptr

Definition at line 377 of file CMS.hh.

6.50.3 Constructor & Destructor Documentation

6.50.3.1 CMS::Transform::Transform (**Profile::ptr** *input,* const **Format** & *informat,* **Profile::ptr** *output,* const **Format** & *outformat,* **Intent** *intent,* cmsUInt32Number *flags*)

Construct a transform from two profiles and formats.

Definition at line 476 of file CMS.cc.

6.50.3.2 CMS::Transform::Transform (std::vector< Profile::ptr > *profile,* const Format & *informat,* const Format & *outformat,* Intent *intent,* cmsUInt32Number *flags*)

Construct a transform from multiple profiles.

Definition at line 485 of file CMS.cc.

6.50.3.3 CMS::Transform::∼Transform ( )

Deconstructor.

Definition at line 492 of file CMS.cc.

6.50.4 Member Function Documentation

6.50.4.1 void CMS::Transform::change_formats (const Format & *informat,* const Format & *outformat*)

Change the input and output formats.

Definition at line 515 of file CMS.cc.

6.50.4.2 Profile::ptr CMS::Transform::device_link (double *version,* cmsUInt32Number *flags*) const

Create a device link profile from this transform.

Definition at line 519 of file CMS.cc.

6.50.4.3 Format CMS::Transform::input_format (void) const

Get the input format.

Definition at line 507 of file CMS.cc.

6.50.4.4 Format CMS::Transform::output_format (void) const

Get the output format.

Definition at line 511 of file CMS.cc.

6.50.4.5 Transform::ptr CMS::Transform::Proofing (Profile::ptr *input,* const Format & *informat,* Profile::ptr *output,* const Format & *outformat,* Profile::ptr *proofing,* Intent *intent,* Intent *proofing_intent,* cmsUInt32Number *flags*) [static]

Named constructor for creating a proofing transform.

Definition at line 496 of file CMS.cc.

6.50.4.6 void CMS::Transform::transform_buffer (const void ∗ *input,* void ∗ *output,* cmsUInt32Number *size*) const

Definition at line 523 of file CMS.cc.

6.50.5 Friends And Related Function Documentation

6.50.5.1 friend class __gnu_cxx::new_allocator< Transform > [friend]

Definition at line 361 of file CMS.hh.

The documentation for this class was generated from the following files:

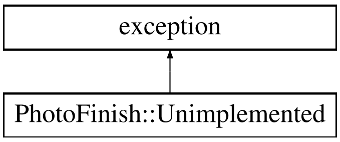
- CMS.hh
- CMS.cc

6.51 PhotoFinish::Unimplemented Class Reference

Unimplemented method exception.

`#include <Exception.hh>`

Inheritance diagram for PhotoFinish::Unimplemented:



Public Member Functions

- Unimplemented (const std::string &c, const std::string &m)

  Constructor.
- virtual const char ∗ what () const throw ()

Protected Attributes

- const std::string _class
- const std::string _method

6.51.1 Detailed Description

Unimplemented method exception.

Definition at line 58 of file Exception.hh.

6.51.2 Constructor & Destructor Documentation

6.51.2.1 PhotoFinish::Unimplemented::Unimplemented (const std::string & *c*, const std::string & *m*) [inline]

Constructor.

Parameters

c	Class name
m	Method name

Definition at line 68 of file Exception.hh.

6.51.3 Member Function Documentation

6.51.3.1 virtual const char∗ PhotoFinish::Unimplemented::what ( ) const throw) [inline], [virtual]

Definition at line 72 of file Exception.hh.

6.51.4 Member Data Documentation

6.51.4.1 const std::string PhotoFinish::Unimplemented::_class [protected]

Definition at line 60 of file Exception.hh.

6.51.4.2 const std::string PhotoFinish::Unimplemented::_method [protected]

Definition at line 60 of file Exception.hh.

The documentation for this class was generated from the following file:

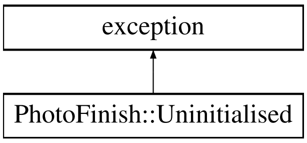
- Exception.hh

6.52 PhotoFinish::Uninitialised Class Reference

Uninitialised attribute exception.

`#include <Exception.hh>`

Inheritance diagram for PhotoFinish::Uninitialised:



Public Member Functions

- Uninitialised (const std::string &c, const std::string &a)

  Constructor.
- Uninitialised (const std::string &c)

  Constructor.
- virtual const char ∗ what () const throw ()

Protected Attributes

- const std::string _class
- const std::string _attribute

6.52.1 Detailed Description

Uninitialised attribute exception.

Definition at line 27 of file Exception.hh.

6.52.2 Constructor & Destructor Documentation

6.52.2.1 PhotoFinish::Uninitialised::Uninitialised (const std::string & *c*, const std::string & *a*) [inline]

Constructor.

Parameters

c	Class name
a	Attribute name

Definition at line 37 of file Exception.hh.

6.52.2.2 PhotoFinish::Uninitialised::Uninitialised (const std::string & *c*) [inline]

Constructor.

Parameters

c	Class name

Definition at line 45 of file Exception.hh.

6.52.3 Member Function Documentation

6.52.3.1 virtual const char∗ PhotoFinish::Uninitialised::what ( ) const throw) [inline], [virtual]

Definition at line 49 of file Exception.hh.

6.52.4 Member Data Documentation

6.52.4.1 const std::string PhotoFinish::Uninitialised::_attribute [protected]

Definition at line 29 of file Exception.hh.

6.52.4.2 const std::string PhotoFinish::Uninitialised::_class [protected]

Definition at line 29 of file Exception.hh.

The documentation for this class was generated from the following file:

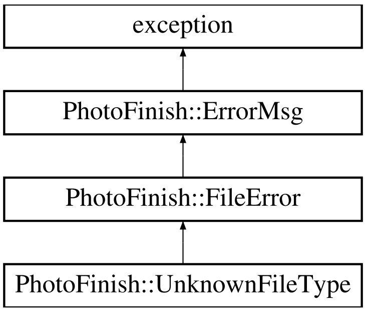
- Exception.hh

6.53 PhotoFinish::UnknownFileType Class Reference

Unknown file type exception.

`#include <Exception.hh>`

Inheritance diagram for PhotoFinish::UnknownFileType:



Public Member Functions

- UnknownFileType (const std::string &fp, const std::string &m)

  Constructor.
- UnknownFileType (const std::string &fp)

  Constructor.
- virtual const char $*$ what () const throw ()

Additional Inherited Members

6.53.1 Detailed Description

Unknown file type exception.

Definition at line 176 of file Exception.hh.

6.53.2 Constructor & Destructor Documentation

6.53.2.1 PhotoFinish::UnknownFileType::UnknownFileType (const std::string & *fp*, const std::string & *m*) [inline]

Constructor.

Parameters

fp	File path
m	Error message

Definition at line 183 of file Exception.hh.

6.53.2.2 PhotoFinish::UnknownFileType::UnknownFileType (const std::string & *fp*) [inline]

Constructor.

Parameters

fp	File path

Definition at line 191 of file Exception.hh.

6.53.3 Member Function Documentation

6.53.3.1 virtual const char$*$ PhotoFinish::UnknownFileType::what ( ) const throw) [inline], [virtual]

Implements PhotoFinish::FileError.

Definition at line 195 of file Exception.hh.

The documentation for this class was generated from the following file:

- Exception.hh

6.54 PhotoFinish::webp_stream_writer Class Reference

A custom writer for libwebp that writes using a std::ostream object.

`#include <WebP_ostream.hh>`

Public Member Functions

- webp_stream_writer (std::ostream ∗s, unsigned int w, unsigned int h)

  Constructor.
- ∼webp_stream_writer ()
- void add_icc (CMS::Profile::ptr profile)

  Add an LCMS2 profile to be written.
- void add_exif (const Exiv2::ExifData &exif)

  Add a set of EXIF tags to be written.
- void add_xmp (const Exiv2::XmpData &xmp)

  Add a set of XMP tags to be written.
- void write_chunk (const char ∗fourcc, const void ∗data, unsigned int length)

  Write a RIFF chunk.
- void before_chunk (void)

  Write stuff before a chunk is written.
- void modify_chunk (unsigned char ∗data)

  Modify the current chunk.
- void after_chunk (void)

  Write stuff after a chunk has been written.
- void modify_vp8x (unsigned char ∗data)
- int write (unsigned char ∗data, size_t data_size)

  Write a block of data from the encoder.

6.54.1 Detailed Description

A custom writer for libwebp that writes using a std::ostream object.

This class is so large because libwebp does not handle metadata *at all*. So we have to keep track of RIFF chunks as the encoder emits them and insert our own, even modifying one of the chunks (VP8X).

Definition at line 34 of file WebP_ostream.hh.

6.54.2 Constructor & Destructor Documentation

6.54.2.1 PhotoFinish::webp_stream_writer::webp_stream_writer (std::ostream ∗ *s*, unsigned int *w*, unsigned int *h*)

Constructor.

Parameters

s	Pointer to a std::ostream derivative.
w,h	Width and height of the image

Definition at line 24 of file WebP_ostream.cc.

6.54.2.2 PhotoFinish::webp_stream_writer::∼webp_stream_writer ( )

Definition at line 32 of file WebP_ostream.cc.

6.54.3 Member Function Documentation

6.54.3.1 void PhotoFinish::webp_stream_writer::add_exif (const Exiv2::ExifData & *exif*)

Add a set of EXIF tags to be written.

Definition at line 48 of file WebP_ostream.cc.

6.54.3.2 void PhotoFinish::webp_stream_writer::add_icc (CMS::Profile::ptr *profile*)

Add an LCMS2 profile to be written.

Definition at line 42 of file WebP_ostream.cc.

6.54.3.3 void PhotoFinish::webp_stream_writer::add_xmp (const Exiv2::XmpData & *xmp*)

Add a set of XMP tags to be written.

Definition at line 60 of file WebP_ostream.cc.

6.54.3.4 void PhotoFinish::webp_stream_writer::after_chunk (void)

Write stuff after a chunk has been written.

Definition at line 107 of file WebP_ostream.cc.

6.54.3.5 void PhotoFinish::webp_stream_writer::before_chunk (void)

Write stuff before a chunk is written.

Definition at line 81 of file WebP_ostream.cc.

6.54.3.6 void PhotoFinish::webp_stream_writer::modify_chunk (unsigned char ∗ *data*)

Modify the current chunk.

Definition at line 99 of file WebP_ostream.cc.

6.54.3.7 void PhotoFinish::webp_stream_writer::modify_vp8x (unsigned char ∗ *data*)

Definition at line 124 of file WebP_ostream.cc.

6.54.3.8 int PhotoFinish::webp_stream_writer::write (unsigned char ∗ *data,* size_t *data_size*)

Write a block of data from the encoder.

Definition at line 136 of file WebP_ostream.cc.

6.54.3.9 void PhotoFinish::webp_stream_writer::write_chunk (const char ∗ *fourcc,* const void ∗ *data,* unsigned int *length*)

Write a RIFF chunk.

Definition at line 71 of file WebP_ostream.cc.

The documentation for this class was generated from the following files:

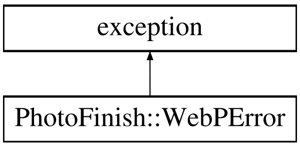
- WebP_ostream.hh
- WebP_ostream.cc

6.55 PhotoFinish::WebPError Class Reference

WebP exception.

`#include <Exception.hh>`

Inheritance diagram for PhotoFinish::WebPError:



Public Member Functions

- WebPError (int c)

  Constructor.
- virtual const char ∗ what () const throw ()

6.55.1 Detailed Description

WebP exception.

Definition at line 322 of file Exception.hh.

6.55.2 Constructor & Destructor Documentation

6.55.2.1 PhotoFinish::WebPError::WebPError (int *c*) [inline]

Constructor.

Parameters

c	Error code

Definition at line 331 of file Exception.hh.

6.55.3 Member Function Documentation

6.55.3.1 virtual const char∗ PhotoFinish::WebPError::what ( ) const throw) [inline], [virtual]

Definition at line 335 of file Exception.hh.

The documentation for this class was generated from the following file:

- Exception.hh

Chapter 7

File Documentation

7.1 CMS.cc File Reference

```
#include <fstream>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <string.h>
#include "CMS.hh"
```

Namespaces

- CMS

Macros

- #define FLOAT_MASK (0xffffffff ^ FLOAT_SH(1))
- #define OPTIMIZED_MASK (0xffffffff ^ OPTIMIZED_SH(1))
- #define COLORSPACE_MASK (0xffffffff ^ COLORSPACE_SH(31))
- #define SWAPFIRST_MASK (0xffffffff ^ SWAPFIRST_SH(1))
- #define FLAVOR_MASK (0xffffffff ^ FLAVOR_SH(1))
- #define PLANAR_MASK (0xffffffff ^ PLANAR_SH(1))
- #define ENDIAN16_MASK (0xffffffff ^ ENDIAN16_SH(1))
- #define DOSWAP_MASK (0xffffffff ^ DOSWAP_SH(1))
- #define EXTRA_MASK (0xffffffff ^ EXTRA_SH(7))
- #define CHANNELS_MASK (0xffffffff ^ CHANNELS_SH(15))
- #define BYTES_MASK (0xffffffff ^ BYTES_SH(7))

Functions

- std::ostream & CMS::operator<< (std::ostream &out, ColourModel model)
- std::ostream & CMS::operator<< (std::ostream &out, Format f)
- cmsIOHANDLER ∗ CMS::OpenIOhandlerFromIStream (std::istream ∗is)
- cmsIOHANDLER ∗ CMS::OpenIOhandlerFromIFStream (fs::path filepath)
- cmsUInt32Number CMS::istream_read (cmsIOHANDLER ∗iohandler, void ∗Buffer, cmsUInt32Number size, cmsUInt32Number count)
- cmsBool CMS::istream_seek (cmsIOHANDLER ∗iohandler, cmsUInt32Number offset)
- cmsBool CMS::istream_close (cmsIOHANDLER ∗iohandler)
- cmsUInt32Number CMS::istream_tell (cmsIOHANDLER ∗iohandler)

- cmsBool CMS::istream_write (cmsIOHANDLER ∗iohandler, cmsUInt32Number size, const void ∗Buffer)
- cmsUInt32Number CMS::ostream_read (cmsIOHANDLER ∗iohandler, void ∗Buffer, cmsUInt32Number size, cmsUInt32Number count)
- cmsBool CMS::ostream_seek (cmsIOHANDLER ∗iohandler, cmsUInt32Number offset)
- cmsBool CMS::ostream_close (cmsIOHANDLER ∗iohandler)
- cmsUInt32Number CMS::ostream_tell (cmsIOHANDLER ∗iohandler)
- cmsBool CMS::ostream_write (cmsIOHANDLER ∗iohandler, cmsUInt32Number size, const void ∗Buffer)
- void lcms2_errorhandler (cmsContext ContextID, cmsUInt32Number ErrorCode, const char ∗Text)

  Throw a LibraryError exception whem LCMS2 returns an error.
- void lcms2_error_adaptor (void)

  Set up an error handler with LCMS2 that will throw a LibraryError exception.

7.1.1 Macro Definition Documentation

7.1.1.1 #define BYTES_MASK (0xffffffff ^ BYTES_SH(7))

Definition at line 250 of file CMS.cc.

7.1.1.2 #define CHANNELS_MASK (0xffffffff ^ CHANNELS_SH(15))

Definition at line 249 of file CMS.cc.

7.1.1.3 #define COLORSPACE_MASK (0xffffffff ^ COLORSPACE_SH(31))

Definition at line 242 of file CMS.cc.

7.1.1.4 #define DOSWAP_MASK (0xffffffff ^ DOSWAP_SH(1))

Definition at line 247 of file CMS.cc.

7.1.1.5 #define ENDIAN16_MASK (0xffffffff ^ ENDIAN16_SH(1))

Definition at line 246 of file CMS.cc.

7.1.1.6 #define EXTRA_MASK (0xffffffff ^ EXTRA_SH(7))

Definition at line 248 of file CMS.cc.

7.1.1.7 #define FLAVOR_MASK (0xffffffff ^ FLAVOR_SH(1))

Definition at line 244 of file CMS.cc.

7.1.1.8 #define FLOAT_MASK (0xffffffff ^ FLOAT_SH(1))

Definition at line 240 of file CMS.cc.

7.1.1.9 #define OPTIMIZED_MASK (0xffffffff ^ OPTIMIZED_SH(1))

Definition at line 241 of file CMS.cc.

7.1.1.10 #define PLANAR_MASK (0xffffffff ^ PLANAR_SH(1))

Definition at line 245 of file CMS.cc.

7.1.1.11 #define SWAPFIRST_MASK (0xffffffff ^ SWAPFIRST_SH(1))

Definition at line 243 of file CMS.cc.

7.1.2 Function Documentation

7.1.2.1 void lcms2_error_adaptor (void)

Set up an error handler with LCMS2 that will throw a LibraryError exception.

Definition at line 641 of file CMS.cc.

7.1.2.2 void lcms2_errorhandler (cmsContext *ContextID,* cmsUInt32Number *ErrorCode,* const char ∗ *Text*)

Throw a LibraryError exception whem LCMS2 returns an error.

Definition at line 637 of file CMS.cc.

7.2 CMS.hh File Reference

```
#include <istream>
#include <ostream>
#include <memory>
#include <boost/filesystem.hpp>
#include <lcms2.h>
#include <lcms2_plugin.h>
#include "Exception.hh"
```

Classes

- class CMS::Profile

  Wrap LCMS2's cmsHPROFILE.
- class CMS::Format

  Wrap LCMS2's pixel format.
- class CMS::Transform

  Wrap LCMS2's transform object.

Namespaces

- CMS

Enumerations

- enum CMS::ColourModel {
  CMS::ColourModel::Any = 0, CMS::ColourModel::Greyscale = 3, CMS::ColourModel::RGB, CMS::Colour-

Model::CMY,
CMS::ColourModel::CMYK, CMS::ColourModel::YCbCr, CMS::ColourModel::YUV, CMS::ColourModel::XYZ, CMS::ColourModel::Lab, CMS::ColourModel::YUVK, CMS::ColourModel::HSV, CMS::ColourModel::HLS, CMS::ColourModel::Yxy, CMS::ColourModel::MCH1, CMS::ColourModel::MCH2, CMS::ColourModel::MCH3,
CMS::ColourModel::MCH4, CMS::ColourModel::MCH5, CMS::ColourModel::MCH6, CMS::ColourModel::MCH7,
CMS::ColourModel::MCH8, CMS::ColourModel::MCH9, CMS::ColourModel::MCH10, CMS::ColourModel::MCH11,
CMS::ColourModel::MCH12, CMS::ColourModel::MCH13, CMS::ColourModel::MCH14, CMS::ColourModel::MCH15,
CMS::ColourModel::LabV2 }

An enum class of LCMS2's colour models.

- enum CMS::Intent {
CMS::Intent::Perceptual, CMS::Intent::Relative_colormetric, CMS::Intent::Saturation, CMS::Intent::Absolute_colormetric,
CMS::Intent::Preserve_k_only_perceptual = 10, CMS::Intent::Preserve_k_only_relative_colormetric, CMS::Intent::Preserve_k_only_saturation, CMS::Intent::Preserve_k_only_absolute_colormetric,
CMS::Intent::Preserve_k_plane_perceptual, CMS::Intent::Preserve_k_plane_relative_colormetric, CMS::Intent::Preserve_k_plane_saturation, CMS::Intent::Preserve_k_plane_absolute_colormetric }

  Wrap LCMS2's intents.

Functions

- std::ostream & CMS::operator<< (std::ostream &out, ColourModel model)
- std::ostream & CMS::operator<< (std::ostream &out, Format f)
- cmsIOHANDLER ∗ CMS::OpenIOhandlerFromIStream (std::istream ∗is)
- cmsIOHANDLER ∗ CMS::OpenIOhandlerFromIFStream (fs::path filepath)
- cmsUInt32Number CMS::istream_read (cmsIOHANDLER ∗iohandler, void ∗Buffer, cmsUInt32Number size, cmsUInt32Number count)
- cmsBool CMS::istream_seek (cmsIOHANDLER ∗iohandler, cmsUInt32Number offset)
- cmsBool CMS::istream_close (cmsIOHANDLER ∗iohandler)
- cmsUInt32Number CMS::istream_tell (cmsIOHANDLER ∗iohandler)
- cmsBool CMS::istream_write (cmsIOHANDLER ∗iohandler, cmsUInt32Number size, const void ∗Buffer)
- cmsUInt32Number CMS::ostream_read (cmsIOHANDLER ∗iohandler, void ∗Buffer, cmsUInt32Number size, cmsUInt32Number count)
- cmsBool CMS::ostream_seek (cmsIOHANDLER ∗iohandler, cmsUInt32Number offset)
- cmsBool CMS::ostream_close (cmsIOHANDLER ∗iohandler)
- cmsUInt32Number CMS::ostream_tell (cmsIOHANDLER ∗iohandler)
- cmsBool CMS::ostream_write (cmsIOHANDLER ∗iohandler, cmsUInt32Number size, const void ∗Buffer)
- void lcms2_error_adaptor (void)

  Set up an error handler with LCMS2 that will throw a LibraryError exception.

7.2.1 Function Documentation

7.2.1.1 void lcms2_error_adaptor (void)

Set up an error handler with LCMS2 that will throw a LibraryError exception.

Definition at line 641 of file CMS.cc.

7.3 CropSolution.cc File Reference

```
#include <boost/lexical_cast.hpp>
#include <math.h>
#include <omp.h>
#include "CropSolution.hh"
```

Namespaces

- PhotoFinish

Macros

- #define sqr(x) ((x) ∗ (x))
- #define min(a, b) ((a) < (b) ? (a) : (b))
- #define max(a, b) ((a) > (b) ? (a) : (b))

Functions

- void PhotoFinish::add_rulers (multihash &vars, std::string key, rulerlist &rulers)

  Parse named variables into a list of rulers.
- void PhotoFinish::add_ruler_pins (rulerlist &rulers, unsigned int max)

  Add rulers to the either side of an image if there aren't enough.

7.3.1 Macro Definition Documentation

7.3.1.1 #define max(*a*, *b*) ((a) > (b) ? (a) : (b))

Definition at line 26 of file CropSolution.cc.

7.3.1.2 #define min(*a*, *b*) ((a) < (b) ? (a) : (b))

Definition at line 25 of file CropSolution.cc.

7.3.1.3 #define sqr(*x*) ((x) ∗ (x))

Definition at line 24 of file CropSolution.cc.

7.4 CropSolution.hh File Reference

```
#include <map>
#include <memory>
#include <ostream>
#include <string>
#include <utility>
#include <list>
#include "Frame.hh"
```

Classes

- class PhotoFinish::CropSolver

  Class for finding the best frame position for cropping.

Namespaces

- PhotoFinish

Typedefs

- typedef std::pair< double, double > PhotoFinish::rulerpair

  Ruler paramaters - percentage of final image vs. pixel position in original.
- typedef std::list< rulerpair > PhotoFinish::rulerlist

  A list of rulers.

7.5 Definable.hh File Reference

```
#include <ostream>
#include <string>
```

Classes

- class PhotoFinish::definable< T >

  Template class for storing things that can be defined or undefined.
- class PhotoFinish::Role_Definable

  Base class for adding "definable" attribute.

Namespaces

- PhotoFinish

7.6 Destination.cc File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <memory>
#include <boost/lexical_cast.hpp>
#include <boost/algorithm/string.hpp>
#include <string.h>
#include "Destination_items.hh"
#include "Destination.hh"
#include "CropSolution.hh"
#include "ImageFile.hh"
#include "Exception.hh"
```

Namespaces

- PhotoFinish

7.7 Destination.hh File Reference

```
#include "yaml-cpp/yaml.h"
#include <string>
#include <map>
#include <boost/filesystem.hpp>
#include "CMS.hh"
#include "Destination_items.hh"
#include "Image.hh"
#include "Frame.hh"
#include "Definable.hh"
```

Classes

- class PhotoFinish::Destination

  Represents a destination, read from destinations.yml.
- class PhotoFinish::Destinations

  A wrapper class for reading destinations from a YAML file and storing them in a map.

Namespaces

- PhotoFinish

7.8 Destination_items.cc File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <boost/lexical_cast.hpp>
#include <boost/algorithm/string.hpp>
#include <string.h>
#include "Destination_items.hh"
#include "Destination.hh"
#include "CropSolution.hh"
#include "Exception.hh"
```

Namespaces

- PhotoFinish

7.9 Destination_items.hh File Reference

```
#include <string>
#include <memory>
#include <vector>
#include "yaml-cpp/yaml.h"
#include <boost/filesystem.hpp>
#include "CMS.hh"
#include "Image.hh"
#include "Definable.hh"
```

Classes

- class PhotoFinish::D_sharpen

  Sharpen parameters for destination.
- class PhotoFinish::D_resize

  Resize parameters for destination.
- class PhotoFinish::D_target

  Target parameters for destination.
- class PhotoFinish::D_JPEG

  JPEG parameters for destination.
- class PhotoFinish::D_PNG

  PNG parameters for destination.
- class PhotoFinish::D_TIFF

  TIFF parameters for destination.
- class PhotoFinish::D_JP2

  JP2 parameters for destination.
- class PhotoFinish::D_WebP

  WebP parameters for destination.
- class PhotoFinish::D_profile

  ICC profile parameters for destination.
- class PhotoFinish::D_thumbnail

  Thumbnail parameters for destination.

Namespaces

- PhotoFinish

Typedefs

- typedef std::map< std::string,
  std::string > PhotoFinish::hash

  A simple hash.
- typedef std::vector< std::string > PhotoFinish::stringlist

  A list of strings.
- typedef std::map< std::string,
  stringlist > PhotoFinish::multihash

  A hash of string lists.

7.10 Ditherer.cc File Reference

```
#include <vector>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "Ditherer.hh"
#include "sample.h"
```

Namespaces

- PhotoFinish

Macros

- #define pos ((x ∗ _channels) + c)
- #define prevpos (((x - 1) ∗ _channels) + c)
- #define nextpos (((x + 1) ∗ _channels) + c)

7.10.1 Macro Definition Documentation

7.10.1.1 #define nextpos (((x + 1) ∗ _channels) + c)

Definition at line 79 of file Ditherer.cc.

7.10.1.2 #define pos ((x ∗ _channels) + c)

Definition at line 77 of file Ditherer.cc.

7.10.1.3 #define prevpos (((x - 1) ∗ _channels) + c)

Definition at line 78 of file Ditherer.cc.

7.11 Ditherer.hh File Reference

```
#include <lcms2.h>
#include "sample.h"
```

Classes

- class PhotoFinish::Ditherer

  Class for dithering images down to 8-bit components.

Namespaces

- PhotoFinish

7.12 Exception.hh File Reference

```
#include <string>
#include <exception>
```

Classes

- class PhotoFinish::Uninitialised

  Uninitialised attribute exception.
- class PhotoFinish::Unimplemented

  Unimplemented method exception.
- class PhotoFinish::NoResults

  No results exception.
- class PhotoFinish::NoTargets

  No targets exception.
- class PhotoFinish::ErrorMsg

  Generic error message exception.
- class PhotoFinish::MemAllocError

  Memory allocation exception.
- class PhotoFinish::FileError

  File error abstract base exception.
- class PhotoFinish::UnknownFileType

  Unknown file type exception.
- class PhotoFinish::FileOpenError

  File open exception.
- class PhotoFinish::FileContentError

  File content exception.
- class PhotoFinish::DestinationError

  Destination exception.
- class PhotoFinish::LibraryError

  Library exception.
- class PhotoFinish::cmsTypeError
- class PhotoFinish::WebPError

  WebP exception.

Namespaces

- PhotoFinish

7.13 Frame.cc File Reference

```
#include <iostream>
#include <iomanip>
#include <omp.h>
#include "Frame.hh"
#include "Destination_items.hh"
#include "Kernel1Dvar.hh"
```

Namespaces

- PhotoFinish

7.14 Frame.hh File Reference

```
#include <memory>
#include "Destination_items.hh"
```

Classes

- class PhotoFinish::Frame

  Crop+rescaling parameters.

Namespaces

- PhotoFinish

7.15 Image.cc File Reference

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <omp.h>
#include "Image.hh"
#include "ImageFile.hh"
```

Namespaces

- PhotoFinish

Functions

- template<typename A , typename B >
  void PhotoFinish::transfer_alpha_typed2 (unsigned int width, unsigned char src_channels, const A ∗src_row, unsigned char dest_channels, const B ∗dest_row)
- template<typename A >
  void PhotoFinish::transfer_alpha_typed (unsigned int width, unsigned char src_channels, const A ∗src_row, CMS::Format dest_format, const void ∗dest_row)
- void PhotoFinish::transfer_alpha (unsigned int width, CMS::Format src_format, const void ∗src_row, CMS::-Format dest_format, const void ∗dest_row)
- std::string PhotoFinish::profile_name (CMS::Profile::ptr profile)

7.16 Image.hh File Reference

```
#include <memory>
#include <exiv2/exiv2.hpp>
#include "Definable.hh"
#include "CMS.hh"
#include "sample.h"
```

Classes

- class PhotoFinish::Image

  An image class.

Namespaces

- PhotoFinish

Functions

- template<typename T >
  T PhotoFinish::scaleval (void)

  A template function that returns the 'scale' value of a type.
- template<>
  unsigned char PhotoFinish::scaleval< unsigned char > (void)
- template<>
  unsigned short int PhotoFinish::scaleval< unsigned short int > (void)
- template<>
  unsigned int PhotoFinish::scaleval< unsigned int > (void)
- template<>
  unsigned long int PhotoFinish::scaleval< unsigned long int > (void)
- template<>
  float PhotoFinish::scaleval< float > (void)
- template<>
  double PhotoFinish::scaleval< double > (void)
- template<typename T >
  T PhotoFinish::limitval (SAMPLE v)

  A template function that limits a floating-point value while converting to another type.
- template<>
  unsigned char PhotoFinish::limitval< unsigned char > (SAMPLE v)
- template<>
  unsigned short int PhotoFinish::limitval< unsigned short int > (SAMPLE v)
- template<>
  unsigned int PhotoFinish::limitval< unsigned int > (SAMPLE v)
- template<>
  unsigned long int PhotoFinish::limitval< unsigned long int > (SAMPLE v)
- template<>
  float PhotoFinish::limitval< float > (SAMPLE v)
- template<>
  double PhotoFinish::limitval< double > (SAMPLE v)

7.17 ImageFile.cc File Reference

```
#include <boost/algorithm/string.hpp>
#include <boost/filesystem.hpp>
#include "ImageFile.hh"
#include "Exception.hh"
```

Namespaces

- PhotoFinish

7.18 ImageFile.hh File Reference

```
#include <string>
#include <memory>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include "CMS.hh"
#include "Image.hh"
#include "Destination.hh"
#include "Exception.hh"
#include "sample.h"
```

Classes

- class PhotoFinish::ImageFilepath

  Class for holding filename and the image format.
- class PhotoFinish::ImageReader

  Abstract base class for reading image files.
- class PhotoFinish::ImageWriter

  Abstract base class for writing image files.
- class PhotoFinish::SOLwriter

  Write the boot logo files for use on Motorola Atrix 4G and possibly other phones.

Namespaces

- PhotoFinish

Functions

- bool PhotoFinish::exists (const ImageFilepath &fp)
- std::time_t PhotoFinish::last_write_time (const ImageFilepath &fp)

7.19 JP2.hh File Reference

```
#include <openjpeg.h>
```

Namespaces

- PhotoFinish

Functions

- void PhotoFinish::error_callback (const char ∗msg, void ∗client_data)

  Error callback for OpenJPEG - throw a LibraryError exception.
- void PhotoFinish::warning_callback (const char ∗msg, void ∗client_data)

  Warning callback for OpenJPEG - print the message to STDERR.
- void PhotoFinish::info_callback (const char ∗msg, void ∗client_data)

  Info callback for OpenJPEG - print the indented message to STDERR.
- template<typename T >
  void PhotoFinish::read_planar (unsigned int width, unsigned char channels, opj_image_t ∗image, T ∗row, unsigned int y)

  Read a row of image data from OpenJPEG's planar integer components into an LCMS2-compatible single array.
- template<typename T >
  void PhotoFinish::write_planar (unsigned int width, unsigned char channels, T ∗row, opj_image_t ∗image, unsigned int y)

  Read a row of planar pixel data into OpenJPEG's planar components.
- template<typename T >
  void PhotoFinish::write_packed (unsigned int width, unsigned char channels, T ∗row, opj_image_t ∗image, unsigned int y)

  Read a row of packed pixel data into OpenJPEG's planar components.

7.20 JP2_callbacks.cc File Reference

```
#include <iostream>
#include <string.h>
#include "Exception.hh"
```

Namespaces

- PhotoFinish

Functions

- void PhotoFinish::error_callback (const char ∗msg, void ∗client_data)

  Error callback for OpenJPEG - throw a LibraryError exception.
- void PhotoFinish::warning_callback (const char ∗msg, void ∗client_data)

  Warning callback for OpenJPEG - print the message to STDERR.
- void PhotoFinish::info_callback (const char ∗msg, void ∗client_data)

  Info callback for OpenJPEG - print the indented message to STDERR.

7.21 JP2reader.cc File Reference

```
#include <boost/algorithm/string.hpp>
#include <openjpeg.h>
#include <omp.h>
#include "ImageFile.hh"
#include "Exception.hh"
#include "JP2.hh"
```

Namespaces

- PhotoFinish

7.22 JP2writer.cc File Reference

```
#include <boost/algorithm/string.hpp>
#include <openjpeg.h>
#include <omp.h>
#include "ImageFile.hh"
#include "Exception.hh"
#include "JP2.hh"
```

Namespaces

- PhotoFinish

7.23 JPEG.hh File Reference

```
#include <stdio.h>
#include <jpeglib.h>
#include "CMS.hh"
#include "Image.hh"
#include "Destination.hh"
```

Namespaces

- PhotoFinish

Functions

- void PhotoFinish::jpeg_istream_src (j_decompress_ptr dinfo, std::istream ∗is)

  Set up a "source manager" on the given JPEG decompression structure to read from an istream.
- void PhotoFinish::jpeg_istream_src_free (j_decompress_ptr dinfo)

  Free the data structures of the istream source manager.
- void PhotoFinish::jpeg_ostream_dest (j_compress_ptr cinfo, std::ostream ∗os)

  Setup a "destination manager" on the given JPEG compression structure to write to an ostream.
- void PhotoFinish::jpeg_ostream_dest_free (j_compress_ptr cinfo)

Free the data structures of the ostream destination manager.

- void PhotoFinish::jpegfile_scan_RGB (jpeg_compress_struct ∗cinfo)

  Create a scan "script" for an RGB image.
- void PhotoFinish::jpegfile_scan_greyscale (jpeg_compress_struct ∗cinfo)

  Create a scan "script" for a greyscale image.
- CMS::Profile::ptr PhotoFinish::jpeg_read_profile (jpeg_decompress_struct ∗dinfo, Destination::ptr dest)

  Read an ICC profile from APP2 markers in a JPEG file.
- void PhotoFinish::jpeg_write_profile (jpeg_compress_struct ∗cinfo, unsigned char ∗data, unsigned int size)

  Write an ICC profile into APP2 markers in a JPEG file.

7.24 JPEG_iostream.cc File Reference

```
#include <iostream>
#include <stdio.h>
#include <jpeglib.h>
#include "Exception.hh"
```

Classes

- struct PhotoFinish::jpeg_source_state_t

  Structure holding information for the istream reader.
- struct PhotoFinish::jpeg_destination_state_t

  Structure holding information for the ostream writer.

Namespaces

- PhotoFinish

Functions

- void PhotoFinish::jpeg_istream_init_source (j_decompress_ptr dinfo)

  Initialise the istream source manager.
- boolean PhotoFinish::jpeg_istream_fill_input_buffer (j_decompress_ptr dinfo)

  Fill the buffer.
- void PhotoFinish::jpeg_istream_skip_input_data (j_decompress_ptr dinfo, long num_bytes)

  Skip some data.
- boolean PhotoFinish::jpeg_istream_resync_to_restart (j_decompress_ptr dinfo, int desired)

  Resync to start?!?
- void PhotoFinish::jpeg_istream_term_source (j_decompress_ptr dinfo)

  Terminate the istream source manager.
- void PhotoFinish::jpeg_istream_src (j_decompress_ptr dinfo, std::istream ∗is)

  Set up a "source manager" on the given JPEG decompression structure to read from an istream.
- void PhotoFinish::jpeg_istream_src_free (j_decompress_ptr dinfo)

  Free the data structures of the istream source manager.
- void PhotoFinish::jpeg_ostream_dest (j_compress_ptr cinfo, std::ostream ∗os)

  Setup a "destination manager" on the given JPEG compression structure to write to an ostream.
- void PhotoFinish::jpeg_ostream_dest_free (j_compress_ptr cinfo)

  Free the data structures of the ostream destination manager.

7.25 JPEG_metadata.cc File Reference

```
#include <exiv2/exiv2.hpp>
#include "JPEG.hh"
```

Namespaces

- PhotoFinish

Functions

- short unsigned int PhotoFinish::read_be16 (const unsigned char ∗data)
- unsigned int PhotoFinish::read_be32 (const unsigned char ∗data)
- void PhotoFinish::jpeg_read_metadata (jpeg_decompress_struct ∗dinfo, Image::ptr img)

7.26 JPEG_profiles.cc File Reference

```
#include <iostream>
#include <map>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <jpeglib.h>
#include "CMS.hh"
#include "Destination.hh"
```

Namespaces

- PhotoFinish

Functions

- CMS::Profile::ptr PhotoFinish::jpeg_read_profile (jpeg_decompress_struct ∗dinfo, Destination::ptr dest)

  Read an ICC profile from APP2 markers in a JPEG file.
- void PhotoFinish::jpeg_write_profile (jpeg_compress_struct ∗cinfo, unsigned char ∗data, unsigned int size)

  Write an ICC profile into APP2 markers in a JPEG file.

7.27 JPEG_scans.cc File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <jpeglib.h>
```

Namespaces

- PhotoFinish

Functions

- void PhotoFinish::jpegfile_scan_RGB (jpeg_compress_struct ∗cinfo)

  Create a scan "script" for an RGB image.
- void PhotoFinish::jpegfile_scan_greyscale (jpeg_compress_struct ∗cinfo)

  Create a scan "script" for a greyscale image.

7.28 JPEGreader.cc File Reference

```
#include <iostream>
#include <queue>
#include <list>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <string.h>
#include <stdio.h>
#include <jpeglib.h>
#include <setjmp.h>
#include <omp.h>
#include "ImageFile.hh"
#include "Image.hh"
#include "JPEG.hh"
```

Namespaces

- PhotoFinish

7.29 JPEGwriter.cc File Reference

```
#include <iostream>
#include <queue>
#include <list>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <string.h>
#include <stdio.h>
#include <jpeglib.h>
#include <setjmp.h>
#include <omp.h>
#include "ImageFile.hh"
#include "Image.hh"
#include "JPEG.hh"
```

Namespaces

- PhotoFinish

7.30 Kernel1Dvar.cc File Reference

```
#include <iostream>
#include <iomanip>
#include <boost/algorithm/string.hpp>
#include <stdlib.h>
#include <math.h>
#include <omp.h>
#include "Kernel1Dvar.hh"
```

Namespaces

- PhotoFinish

Macros

- #define sqr(x) ((x) $*$ (x))
- #define min(x, y) ((x) $<$ (y) ? (x) : (y))

7.30.1 Macro Definition Documentation

7.30.1.1 #define min(*x,* *y*) ((x) $<$ (y) ? (x) : (y))

Definition at line 28 of file Kernel1Dvar.cc.

7.30.1.2 #define sqr(*x*) ((x) $*$ (x))

Definition at line 27 of file Kernel1Dvar.cc.

7.31 Kernel1Dvar.hh File Reference

```
#include <memory>
#include "Destination_items.hh"
#include "Exception.hh"
#include "Definable.hh"
#include "sample.h"
```

Classes

- class PhotoFinish::Kernel1Dvar

  Creates and stores coefficients for cropping and resizing an image.
- class PhotoFinish::Lanczos

  Lanczos filter.

Namespaces

- PhotoFinish

7.32 Kernel2D.cc File Reference

```
#include <stdlib.h>
#include <omp.h>
#include "Kernel2D.hh"
#include "Destination_items.hh"
```

Namespaces

- PhotoFinish

Macros

- #define sqr(x) ((x) ∗ (x))

7.32.1 Macro Definition Documentation

7.32.1.1 #define sqr(*x*) ((x) ∗ (x))

Definition at line 24 of file Kernel2D.cc.

7.33 Kernel2D.hh File Reference

```
#include <memory>
#include "Image.hh"
#include "Exception.hh"
#include "Definable.hh"
#include "sample.h"
```

Classes

- class PhotoFinish::Kernel2D

  Creates and stores coefficients for convolving an image.
- class PhotoFinish::GaussianSharpen

  GaussianSharpen kernel.

Namespaces

- PhotoFinish

7.34 Kernel2Dvar.cc File Reference

```
#include <iostream>
#include <iomanip>
#include <boost/algorithm/string.hpp>
#include <stdlib.h>
#include <math.h>
#include <omp.h>
#include "Kernel2Dvar.hh"
#include "vector.h"
```

Namespaces

- PhotoFinish

Macros

- #define sqr(x) ((x) $*$ (x))
- #define min(x, y) ((x) $<$ (y) ? (x) : (y))

7.34.1 Macro Definition Documentation

7.34.1.1 #define min(*x*, *y*) ((x) $<$ (y) ? (x) : (y))

Definition at line 29 of file Kernel2Dvar.cc.

7.34.1.2 #define sqr(*x*) ((x) $*$ (x))

Definition at line 28 of file Kernel2Dvar.cc.

7.35 Kernel2Dvar.hh File Reference

```
#include <memory>
#include "Destination_items.hh"
#include "Exception.hh"
#include "Definable.hh"
#include "sample.h"
```

Classes

- class PhotoFinish::Kernel2Dvar

  Creates and stores coefficients for cropping and resizing an image.
- class PhotoFinish::Lanczos2D

  Lanczos filter.

Namespaces

- PhotoFinish

7.36 LCMS2ErrorHandler.cc File Reference

```
#include <lcms2.h>
#include "Exception.hh"
```

Namespaces

- PhotoFinish

Functions

- void PhotoFinish::lcms2_errorhandler (cmsContext ContextID, cmsUInt32Number ErrorCode, const char ∗Text)

  Throw a LibraryError exception whem LCMS2 returns an error.
- void PhotoFinish::lcms2_error_adaptor (void)

  Set up an error handler with LCMS2 that will throw a LibraryError exception.

7.37 photofinish.cc File Reference

```
#include <iostream>
#include <string>
#include <deque>
#include <boost/filesystem.hpp>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include "Image.hh"
#include "ImageFile.hh"
#include "Destination.hh"
#include "Tags.hh"
#include "Kernel2D.hh"
#include "Exception.hh"
```

Functions

- int main (int argc, char ∗argv[])

7.37.1 Function Documentation

7.37.1.1 int main (int *argc,* char ∗ *argv[]*)

Definition at line 37 of file photofinish.cc.

7.38 PNG_metadata.cc File Reference

```
#include <iostream>
#include <zlib.h>
#include "PNG_metadata.hh"
```

Namespaces

- PhotoFinish

Functions

- void PhotoFinish::png_read_metadata (png_structp png, png_infop info, Image::ptr image)

7.39 PNG_metadata.hh File Reference

```
#include <png.h>
#include "Image.hh"
```

Namespaces

- PhotoFinish

Functions

- void PhotoFinish::png_read_metadata (png_structp png, png_infop info, Image::ptr image)

7.40 PNGreader.cc File Reference

```
#include <errno.h>
#include <png.h>
#include <zlib.h>
#include <time.h>
#include <omp.h>
#include <boost/algorithm/string.hpp>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <iostream>
#include "ImageFile.hh"
#include "Image.hh"
#include "PNGreader_cb.hh"
```

Namespaces

- PhotoFinish

7.41 PNGreader_cb.cc File Reference

```
#include "PNGreader_cb.hh"
```

Namespaces

- PhotoFinish

Functions

- void PhotoFinish::png_info_cb (png_structp png, png_infop info)

  Called by libPNG when the iHDR chunk has been read with the main "header" information.
- void PhotoFinish::png_row_cb (png_structp png, png_bytep row_data, png_uint_32 row_num, int pass)

  Called by libPNG when a row of image data has been read.
- void PhotoFinish::png_end_cb (png_structp png, png_infop info)

  Called by libPNG when the image data has finished.

7.42 PNGreader_cb.hh File Reference

```
#include <png.h>
#include "Destination.hh"
#include "Image.hh"
```

Classes

- struct PhotoFinish::PNGreader_cb

Namespaces

- PhotoFinish

Functions

- void PhotoFinish::png_info_cb (png_structp png, png_infop info)

  Called by libPNG when the iHDR chunk has been read with the main "header" information.
- void PhotoFinish::png_row_cb (png_structp png, png_bytep row_data, png_uint_32 row_num, int pass)

  Called by libPNG when a row of image data has been read.
- void PhotoFinish::png_end_cb (png_structp png, png_infop info)

  Called by libPNG when the image data has finished.

7.43 PNGwriter.cc File Reference

```
#include <errno.h>
#include <png.h>
#include <zlib.h>
#include <time.h>
#include <omp.h>
#include <boost/algorithm/string.hpp>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <iostream>
#include "ImageFile.hh"
#include "Image.hh"
```

Namespaces

- PhotoFinish

Functions

- void PhotoFinish::png_write_ostream_cb (png_structp png, png_bytep buffer, png_size_t length)

  libPNG callback for writing to an ostream
- void PhotoFinish::png_flush_ostream_cb (png_structp png)

  libPNG callback for flushing an ostream

7.44 process_scans.cc File Reference

```
#include <boost/program_options/options_description.hpp>
#include <boost/program_options/variables_map.hpp>
#include <boost/program_options/parsers.hpp>
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
#include <boost/filesystem.hpp>
#include <boost/algorithm/string.hpp>
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include "CMS.hh"
#include "Image.hh"
#include "ImageFile.hh"
#include "Destination.hh"
#include "Tags.hh"
#include "Kernel2D.hh"
#include "Exception.hh"
```

Functions

- void make_preview (Image::ptr orig_image, Destination::ptr orig_dest, Tags::ptr filetags, ImageWriter::ptr preview_file, bool can_free=false)
- void preview_dir (fs::path dir, std::string format, std::shared_ptr< Tags > tags)
- int main (int argc, char ∗argv[])

7.44.1 Function Documentation

7.44.1.1 int main (int *argc,* char ∗ *argv[]*)

Definition at line 116 of file process_scans.cc.

7.44.1.2 void make_preview (Image::ptr *orig_image,* Destination::ptr *orig_dest,* Tags::ptr *filetags,* ImageWriter::ptr *preview_file,* bool *can_free* = `false`)

Definition at line 46 of file process_scans.cc.

7.44.1.3 void preview_dir (fs::path *dir,* std::string *format,* std::shared_ptr< Tags > *tags*)

Definition at line 85 of file process_scans.cc.

7.45 sample.h File Reference

Macros

- #define SAMPLE float
- #define SET_SAMPLE_FORMAT(x) ((x).set_float())
- #define SAMPLE_VECTOR f4vector
- #define SAMPLE_VECTOR_SIZE 4

7.45.1 Macro Definition Documentation

7.45.1.1 #define SAMPLE float

Definition at line 20 of file sample.h.

7.45.1.2 #define SAMPLE_VECTOR f4vector

Definition at line 30 of file sample.h.

7.45.1.3 #define SAMPLE_VECTOR_SIZE 4

Definition at line 31 of file sample.h.

7.45.1.4 #define SET_SAMPLE_FORMAT(*x*) ((x).set_float())

Definition at line 26 of file sample.h.

7.46 SOLwriter.cc File Reference

```
#include "ImageFile.hh"
#include "Image.hh"
#include "Ditherer.hh"
```

Namespaces

- PhotoFinish

Functions

- void PhotoFinish::write_be (void ∗ptr, size_t size, std::ostream &stream)

Variables

- unsigned char PhotoFinish::header [12]

7.47 StreamIO.cc File Reference

```
#include "StreamIO.hh"
```

Namespaces

- PhotoFinish

7.48 StreamIO.hh File Reference

```
#include <iostream>
#include <exiv2/exiv2.hpp>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <boost/static_assert.hpp>
#include <boost/type_traits.hpp>
```

Classes

- class PhotoFinish::StreamIO

Namespaces

- PhotoFinish

7.49 Tags.cc File Reference

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <map>
#include <exiv2/exiv2.hpp>
#include <math.h>
#include "Image.hh"
#include "ImageFile.hh"
#include "Tags.hh"
#include "Exception.hh"
```

Namespaces

- PhotoFinish

Functions

- Exiv2::ExifKey PhotoFinish::exif_key_read (std::string key_string)
- Exiv2::Value::AutoPtr PhotoFinish::exif_value_read (Exiv2::ExifKey key, std::string value_string)

Read an EXIF value from a string, with optional substitution for enum-style values.

- Exiv2::IptcKey PhotoFinish::iptc_key_read (std::string key_string)
- Exiv2::XmpKey PhotoFinish::xmp_key_read (std::string key_string)

7.50 Tags.hh File Reference

```
#include <exiv2/exiv2.hpp>
#include <boost/filesystem.hpp>
#include <boost/lexical_cast.hpp>
#include <iostream>
#include <string>
#include <map>
#include <memory>
#include <list>
#include "Image.hh"
#include "Destination.hh"
```

Classes

- class PhotoFinish::Tags

  Reads and holds tag information.

Namespaces

- PhotoFinish

Macros

- #define StrPair(s, v) std::make_pair<std::string, std::string>(s, v)

Typedefs

- typedef std::vector< std::pair
  < std::string, std::string > > PhotoFinish::subst_table

Functions

- template<typename Num_type , typename R_type >
  Exiv2::ValueType< R_type > & PhotoFinish::closest_Rational (double value)

  Find a close rational fraction given a floating-point value.

7.50.1 Macro Definition Documentation

7.50.1.1 #define StrPair(*s,* *v*) std::make_pair<std::string, std::string>(s, v)

Definition at line 38 of file Tags.hh.

7.51 Tags_EXIF_subst.cc File Reference

```
#include <string>
#include <map>
#include <boost/algorithm/string.hpp>
#include "Tags.hh"
```

Namespaces

- PhotoFinish

Macros

- #define Key(k, h) std::make_pair<std::string, subst_table>(k, h)

Functions

- Exiv2::ExifKey PhotoFinish::exif_key_read (std::string key_string)
- template<typename Num_type , typename R_type >
  Exiv2::Value::AutoPtr PhotoFinish::parse_Rational (std::string s)

  Parse a string into a rational fraction.
- Exiv2::Value::AutoPtr PhotoFinish::exif_value_read (Exiv2::ExifKey key, std::string value_string)

  Read an EXIF value from a string, with optional substitution for enum-style values.

Variables

- subst_table PhotoFinish::EXIF_key_subst

  Map from Image::Exiftool tag names to Exiv2's tag names.
- std::map< std::string,
  subst_table > PhotoFinish::EXIF_value_subst

7.51.1 Macro Definition Documentation

7.51.1.1 #define Key(*k*, *h*) std::make_pair<std::string, subst_table>(k, h)

Definition at line 91 of file Tags_EXIF_subst.cc.

7.52 Tags_IPTC_subst.cc File Reference

```
#include <string>
#include <map>
#include <boost/algorithm/string.hpp>
#include "Tags.hh"
```

Namespaces

- PhotoFinish

Functions

- Exiv2::IptcKey PhotoFinish::iptc_key_read (std::string key_string)

Variables

- subst_table PhotoFinish::IPTC_key_subst

  Map from Image::Exiftool tag names to Exiv2's tag names.

7.53 Tags_XMP_subst.cc File Reference

```
#include <string>
#include <map>
#include <boost/algorithm/string.hpp>
#include "Tags.hh"
```

Namespaces

- PhotoFinish

Functions

- Exiv2::XmpKey PhotoFinish::xmp_key_read (std::string key_string)

Variables

- subst_table PhotoFinish::XMP_key_subst

  Map from Image::Exiftool tag names to Exiv2's tag names.

7.54 TIFFreader.cc File Reference

```
#include <boost/algorithm/string.hpp>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <tiffio.h>
#include <tiffio.hxx>
#include "ImageFile.hh"
```

Namespaces

- PhotoFinish

Macros

- #define TIFFcheck(x) if ((rc = TIFF##x) != 1) throw LibraryError("libtiff", "TIFF" #x " returned " + rc)

7.54.1 Macro Definition Documentation

7.54.1.1 #define TIFFcheck(*x*) if ((rc = TIFF##x) != 1) throw LibraryError("libtiff", "TIFF" #x " returned " + rc)

Definition at line 34 of file TIFFreader.cc.

7.55 TIFFwriter.cc File Reference

```
#include <boost/algorithm/string.hpp>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <tiffio.h>
#include <tiffio.hxx>
#include "ImageFile.hh"
```

Namespaces

- PhotoFinish

Macros

- #define TIFFcheck(x) if ((rc = TIFF##x) != 1) throw LibraryError("libtiff", "TIFF" #x " returned " + rc)

7.55.1 Macro Definition Documentation

7.55.1.1 #define TIFFcheck(*x*) if ((rc = TIFF##x) != 1) throw LibraryError("libtiff", "TIFF" #x " returned " + rc)

Definition at line 34 of file TIFFwriter.cc.

7.56 vector.h File Reference

Classes

- union f2vector
- union f4vector
- union d2vector
- union d4vector

Typedefs

- typedef float v2sf __attribute__ ((vector_size(8)))

7.56.1 Typedef Documentation

7.56.1.1 typedef double v4df __attribute__((vector_size(32)))

Definition at line 26 of file vector.h.

7.57 WebP_ostream.cc File Reference

```
#include <webp/encode.h>
#include "WebP_ostream.hh"
```

Namespaces

- PhotoFinish

Macros

- #define min(a, b) ((a) < (b) ? (a) : (b))

Functions

- int PhotoFinish::webp_stream_writer_func (const uint8_t ∗data, size_t data_size, const WebPPicture ∗picture)

    Wrapper around the webp_stream_writer class.

7.57.1 Macro Definition Documentation

7.57.1.1 #define min(*a, b*) ((a) < (b) ? (a) : (b))

Definition at line 133 of file WebP_ostream.cc.

7.58 WebP_ostream.hh File Reference

```
#include <iostream>
#include <exiv2/exiv2.hpp>
#include "CMS.hh"
```

Classes

- class PhotoFinish::webp_stream_writer

    A custom writer for libwebp that writes using a std::ostream object.

Namespaces

- PhotoFinish

Functions

- int PhotoFinish::webp_stream_writer_func (const uint8_t ∗data, size_t data_size, const WebPPicture ∗picture)

    Wrapper around the webp_stream_writer class.
- void PhotoFinish::copy_le_to (unsigned char ∗dest, unsigned int value, unsigned char length)
- unsigned int PhotoFinish::read_le32 (const unsigned char ∗data)

7.59 WebPreader.cc File Reference

```
#include <boost/algorithm/string.hpp>
#include <webp/decode.h>
#include <omp.h>
#include "ImageFile.hh"
#include "Exception.hh"
#include "WebP_ostream.hh"
```

Namespaces

- PhotoFinish

7.60 WebPwriter.cc File Reference

```
#include <boost/algorithm/string.hpp>
#include <webp/encode.h>
#include <omp.h>
#include "ImageFile.hh"
#include "Exception.hh"
#include "WebP_ostream.hh"
```

Namespaces

- PhotoFinish

Variables

- std::map< std::string, WebPPreset > PhotoFinish::WebP_presets

Index