MinImgIO Library Reference

version 1.0

Generated by Doxygen 1.7.5.1

Wed Oct 12 2011 20:22:05

Contents

1 Overview

MinImgIO is an open-source platform-independent library for reading and writing image files. The library does not contain any implementations of encode/decode algorithms. Rather than do this, it uses third party open-source cross-platform libraries. The following table provides a summary of the supported image formats and respective libraries:

Format	Description	Library
TIFF	Tagged Image File Format	libtiff
JPEG	Joint Photographic Experts Group	libjpeg

For the internal representation of images is used cross-platform open-source container - MinImg (see Image Representation section for more information). The advantages of this container are the using a minimal number of fields needed to represent the bitmap image and the easy way to cast it to other standard and popular containers (for instance, Windows DIB, GDI+ BitmapData, Intel/OpenCV IplImage).

MinImgIO library allows write/read images to/from both file system and memory block (see Loading Images from Memory for more information).

The library is written in C++ and can be compiled under Linux (GCC) and Windows (MSVC 8 and later). Though the library has been written in C++, it has C interface, so it can be embedded in different systems.

2 Quick Tutorial

This tutorial is intended to get you start using **MinImgIO** library to simply read and write images, therefore the tutorial is not a complete or detailed documentation of the library. Note also, that some secondary operations will be purposely omitted for brevity.

2.1 Reading and Writing Images

Let `szImagePath` is a null-terminated string that contains the physical path of the image. At the first step we should get image properties (size, channel number, depth and other). The following code shows how to open the image and retrieve the properties of the first page:

```
MinImg image = {0};
PROPAGATE_ERROR(GetMinImageFileProps(&image, szImagePath, 0));
```

Then we will allocate the memory for the image data. To do that we use `AllocMinImage()` function from **MinImgAPI** library:

```
PROPAGATE_ERROR(AllocMinImage(&image, 16));
```

Now we are ready to read image data. The following code demonstrates loading the first page of the image:

```
PROPAGATE_ERROR(LoadMinImage(&image, szImagePath, 0));
```

Now move to writing images. Let we have already some `image` object and want to save it to the file `szImagePath`. The following code shows how to do that:

```
PROPAGATE_ERROR(SaveMinImage(szImagePath, &image, 0));
```

Note that `SaveMinImage()` function determines the proper file format based on the filename extension. If you want to specify it manually, you should use more comprehensive function `SaveMinImageEx()`.

2.2 Loading Images from Memory

Sometimes it is not practical or even possible to load an image from disk. For such situations LoadMinImage() allows to read an image from memory block which contains valid image format. To do this, you should use a special format of filename:

```
mem://<pointer-to-memory-block>.<size-of-memory-block>
```

Let `pImageData` is a pointer to the image in memory and `imageSize` is the size the image. The following example demonstrates a way to generate filename to the image in memory:

```
char szImageMemPath[250] = {0};
sprintf(szImageMemPath, "mem://%p.%lu", pImageData, imageSize);
```

3 MinImgIO License Agreements

3.1 Library License Agreement

MinImgIO is released under FreeBSD License. It is free for both academic and commercial use.

```
Copyright (c) 2011, Smart Engines Limited. All rights reserved.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification
    ,
are permitted provided that the following conditions are met:

   1. Redistributions of source code must retain the above copyright notice,
      this list of conditions and the following disclaimer.

   2. Redistributions in binary form must reproduce the above copyright notice,
      this list of conditions and the following disclaimer in the documentation
      and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESS OR
```

```
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
      EVENT
SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those
of the authors and should not be interpreted as representing official policies,
either expressed or implied, of copyright holders.
```

3.2 Documentation License Agreement

This documentation is released under FreeBSD Documentation License. It is free for both academic and commercial use.

```
Copyright (c) 2011, Smart Engines Limited. All rights reserved.

All rights reserved.

Redistribution and use in source (doxygen documentation blocks) and 'compiled'
forms (HTML, PDF, PostScript, RTF and so forth) with or without modification,
are permitted provided that the following conditions are met:

   1. Redistributions of source code (doxygen documentation blocks) must retain

      the above copyright notice, this list of conditions and the following
      disclaimer as the first lines of this file unmodified.

   2. Redistributions in compiled form (converted to PDF, PostScript, RTF and
      other formats) must reproduce the above copyright notice, this list of
      conditions and the following disclaimer in the documentation and/or other
      materials provided with the distribution.

THIS DOCUMENTATION IS PROVIDED BY COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
      EVENT
SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENTATION, EVEN IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

4 Module Documentation

4.1 MinImgIO Library API

This section describes an application programming interface (API) of **MinImgIO** library. Though **MinImgIO** has been written in C++, it has C interface to make it easy embedding the library in different systems.

Data Structures

- struct ExtImgProps

  Specifies additional information for an image. More...

Defines

- #define IS_BY_DEFAULT(a)

  Specifies a default value for a parameter of a function.
- #define MINIMGIO_API

  Specifies storage-class information (only for MSC).

Enumerations

- enum ImgFileFormat

  Specifies supported file formats.
- enum ImgFileComp

  Specifies supported TIFF compressions.

Functions

- int GuessImageFileFormat (const char *pFileName)

  Detect the image file format by the magic bytes or by the name.
- int GetMinImageFilePages (const char *pFileName)

  Returns the number of pages.
- int GetMinImageFileProps (MinImg *pImg, const char *pFileName, int page)

  Gets basic information about an image.
- int GetMinImageFilePropsEx (MinImg *pImg, ExtImgProps *pProps, const char *pFileName, int page)

  Gets detailed information about an image.
- int LoadMinImage (const MinImg *pImg, const char *pFileName, int page)

  Loads an image from a file.
- int SaveMinImage (const char *pFileName, const MinImg *pImg, int page)

  Saves an image to a specified file.
- int SaveMinImageEx (const char *pFileName, const MinImg *pImg, const ExtImgProps *pProps, int page)

  Saves an image to a specified file with specified options.
- int PackMinImage (const MinImg *pDst, const MinImg *pSrc, uint8_t level)

  Packs a grayscale image into monochrome once.
- int UnpackMinImage (const MinImg *pDst, const MinImg *pSrc)

  Unpacks a monochrome image into grayscale one.

4.1.1 Data Structure Documentation

4.1.1.1 struct ExtImgProps

The structure specifies additional information about the image such as horizontal and vertical DPI, file format and lossy quality. This is used both in input functions (to get additional information about the input image) and output functions (to specify the proper way of writing the image).

Definition at line 236 of file minimgio.h.

Data Fields

- ImgFileFormat iff

  The image file format (see ImgFileFormat).
- ImgFileComp comp

  The image file compression (see ImgFileComp).
- float xDPI

  The horizontal resolution, in dots-per-inch.
- float yDPI

  The vertical resolution, in dots-per-inch.
- int qty

  The resultant image quality (for JPEG compression).

4.1.2 Enumeration Type Documentation

4.1.2.1 enum ImgFileComp

The enum specifies all supported TIFF compressions.

Enumerator:

IFC_NONE No compression.

IFC_LZW Lempel-Ziv & Welch algorithm.

IFC_GROUP4 CCITT Group 4 fax encoding.

Definition at line 220 of file minimgio.h.

4.1.2.2 enum ImgFileFormat

The enum specifies all supported input/output file formats.

Enumerator:

IFF_UNKNOWN Unknown file format.

IFF_TIFF Tagged image file format.

IFF_JPEG JPEG file format.

Definition at line 208 of file minimgio.h.

4.1.3 Function Documentation

4.1.3.1 int GetMinImageFilePages (const char * *pFileName*)

Parameters

pFileName	The filename of the image to get the page count for.

Returns

`NO_ERRORS` on success or an error code otherwise (see `MinErr`).

The function returns the number of pages that are in the specified image file.

4.1.3.2 int GetMinImageFileProps (MinImg * *pImg*, const char * *pFileName*, int *page*)

Parameters

pImg	The image to be filled.
pFileName	The filename of the image to get properties for.
page	0-based page number.

Returns

`NO_ERRORS` on success or an error code otherwise (see `MinErr`).

The function opens the specified image file, extracts the image information and fills appropriate fields of `pImg` (`pImg->width`, `pImg->height`, `pImg->channels`, `pImg->channelDepth`, and `pImg->format`). This also zeros `pImg->pScan0` and `pImg->stride` fields.

4.1.3.3 int GetMinImageFilePropsEx (MinImg * *pImg*, ExtImgProps * *pProps*, const char * *pFileName*, int *page*)

Parameters

pImg	The image to be filled.
pProps	The additional information about the image.
pFileName	The filename of the image to get properties for.
page	0-based page number.

Returns

`NO_ERRORS` on success or an error code otherwise (see `MinErr`).

The function opens the specified image file, extracts the image information and fills appropriate fields of `pImg` (in the same way as the function GetMinImageFileProps()). Moreover, this gets some additional information (such as horizontal and vertical resolutions) and stores it into pProps.

4.1.3.4 int GuessImageFileFormat (const char * *pFileName*)

Parameters

pFileName	The filename of the image to guest the format for.

Returns

One of the available file format (see `ImgFileFormat`) on success or `IFF_UNKNOWN` if the function was unable to guess.

The function opens the image, reads the first bits (magic numbers) of a file which uniquely identify the type of file. If the file does not exist or the function cannot open the file, then it try to guess the format just using the file extension. Currently the following file formats are supported:

- TIFF files - ∗.tiff, ∗.tif
- JPEG files - ∗.jpeg, ∗.jpg

4.1.3.5 int LoadMinImage (const MinImg ∗ *pImg*, const char ∗ *pFileName*, int *page*)

Parameters

pImg	Loaded image.
pFileName	The filename of the image to load.
page	0-based page number.

Returns

`NO_ERRORS` on success or an error code otherwise (see `MinErr`).

The function loads an image from the specified file. The image data are placed into `pImg` which must be allocated in advance. This function automatically detects the format of the image to load.

4.1.3.6 int PackMinImage (const MinImg ∗ *pDst*, const MinImg ∗ *pSrc*, uint8_t *level*)

Parameters

pDst	The output 1-bit single-channel image.
pSrc	The input 8-bit single-channel image.
level	The threshold value.

Returns

`NO_ERRORS` on success or an error code otherwise (see `MinErr`).

The function converts the input grayscale (8-bit) image into monochrome 1-bit one using the specified threshold value. The pixel treats as black if its value is less then threshold and as white otherwise.

4.1.3.7 int SaveMinImage (const char ∗ *pFileName*, const MinImg ∗ *pImg*, int *page*)

Parameters

pFileName	The name of the file to save the image.
pImg	The image to be saved.
page	0-based page number.

Returns

NO_ERRORS on success or an error code otherwise (see MinErr).

The function saves the image to the specified file. The image format is chosen based on the filename extension.

4.1.3.8 int SaveMinImageEx (const char * *pFileName*, const MinImg * *pImg*, const ExtImgProps * *pProps*, int *page*)

Parameters

pFileName	The name of the file to save the image.
pImg	The image to be saved.
pProps	The specified save parameters.
page	0-based page number.

Returns

NO_ERRORS on success or an error code otherwise (see MinErr).

The function saves the image to the specified file. This function also takes into account the additional save parameters pProps, which can be used to specify the image format, resolutions, and the quality. If the image format is not specified by the pProps argument, then it will be chosen based on the filename extension.

4.1.3.9 int UnpackMinImage (const MinImg * *pDst*, const MinImg * *pSrc*)

Parameters

pDst	The output 8-bit single-channel image.
pSrc	The input 1-bit single-channel image.

Returns

NO_ERRORS on success or an error code otherwise (see MinErr).

The function converts the input monochrome 1-bit image into grayscale 8-bit one. Zero is mapped into zero and one is mapped into 255.

4.2 Possible Return Codes

The module specifies the return values used thought the library. Every function in the library follows the rule: it returns integer value. Meanwhile, a nonnegative return value indicates that the function completed successfully whereas a negative value indicates erroneous execution and specifies the error code. The enum `MinErr` contains codes for the most common errors. It is convenient enough to use special defines for handling return codes. Below you can find two defines which are widely used in the library.

Defines

- #define PROPAGATE_ERROR(call)

  If function failed then propagate the error code.
- #define SHOULD_WORK(call)

  If function failed then propagate INTERNAL_ERROR.

Enumerations

- enum MinErr

  Specifies basic error codes.

4.2.1 Define Documentation

4.2.1.1 #define PROPAGATE_ERROR(*call*)

Value:

```
{ \
  int res = call; \
  if (res < 0) \
    return res; \
}
```

This define macro describes a code that helps to propagate an exception if an error occurs.

Definition at line 85 of file minerr.h.

4.2.1.2 #define SHOULD_WORK(*call*)

Value:

```
{ \
  int res = call; \
  if (res < 0) \
    return INTERNAL_ERROR; \
}
```

This define macro describes a code that propagate INTERNAL_ERROR exception if an error occurs.

Definition at line 98 of file minerr.h.

4.2.2 Enumeration Type Documentation

4.2.2.1 enum MinErr

The enum specifies a list of basic error codes that is such ones which can be returned by any function in the library.

Enumerator:

NO_ERRORS No error has occurred. It indicates that the function completed successfully.

BAD_ARGS This error indicates that one or more arguments passed to the function are not correct.

NO_MEMORY Not enough memory is available. This can result from low memory conditions.

NOT_IMPLEMENTED This error indicates that the requested function is not implemented.

INTERNAL_ERROR An internal error has occurred. This error indicates that something went wrong.

FILE_ERROR An error occurred while working with files. The most likely cause is a full disk or a corrupted file to be open.

Definition at line 62 of file minerr.h.

4.3 Image Representation

The module specifies the image representation format. Every function in the library expects an input image in the form of `MinImg` object. `MinImg` is a cross-platform open-source container. The advantages of this container are the using minimal number of fields needed to represent the bitmap image and the easy way to cast it to other standard and popular view (for instance, Windows DIB, GDI+ BitmapData, Intel/OpenCV IplImage).

Data Structures

- struct MinImg

  A low-level universal representation of a bitmap image. More...
- struct float16

  Specifies half-precision floating point. More...

Enumerations

- enum MinFmt

  Specifies acceptable element formats of each individual channel.
- enum MinTyp

  Specifies acceptable element types of each individual channel.

4.3.1 Data Structure Documentation

4.3.1.1 struct MinImg

The struct `MinImg` represents a 2D dense numerical with additional fields needed for image representations (format and channel number). The struct `MinImg` allows to describe single-channel and multi- channel images in a wide range of different image types. Herewith, the format of the image is specified by two values: depth of the channel (see `MinImg::channelDepth`) and channel element format (see `MinImg::format`). To represent a binary image you should set `MinImg::format` to `FMT_UINT` and `MinImg::channelDepth` to 0.

Definition at line 69 of file minimg.h.

Data Fields

- int32_t width

  The image width in pixels. It must be positive.
- int32_t height

  The image height in pixels. It must be positive.
- int32_t stride

  The width of a single row of pixels in bytes.
- int32_t channels

The number of channels per pixel. It must be positive.

- int32_t channelDepth

  The channel depth in bytes. It must be nonnegative.

- MinFmt format

  The channel element format (see `MinFmt`).

- uint8_t * pScan0

  The pointer to the first pixel of the first row.

4.3.1.2 struct float16

The struct `float16` represents half-precision floating point.

Definition at line 90 of file mintyp.h.

Data Fields

- uint16_t significand: 10

  The mantissa of the number.

- uint16_t exponent: 5

  The magnitude of the number.

- uint16_t sign: 1

  The sing of the number.

4.3.2 Enumeration Type Documentation

4.3.2.1 enum MinFmt

The enum specifies acceptable element formats of each individual channel.

Enumerator:

FMT_UINT Unsigned integer.

FMT_INT Signed integer.

FMT_FLOAT Floating point.

Definition at line 56 of file mintyp.h.

4.3.2.2 enum MinTyp

The enum specifies acceptable element types (that is format + size) of each individual channel.

Enumerator:

TYP_UINT1 1-bit logical.

TYP_UINT8 Unsigned 8-bit integer.

TYP_INT8 Signed 8-bit integer.

TYP_UINT16 Unsigned 16-bit integer.

TYP_INT16 Signed 16-bit integer.

TYP_FLOAT16 Half-precision floating point.

TYP_UINT32 Unsigned 32-bit integer.

TYP_INT32 Signed 32-bit integer.

TYP_FLOAT32 Single-precision floating point.

TYP_UINT64 Unsigned 64-bit integer.

TYP_INT64 Signed 64-bit integer.

TYP_FLOAT64 Double-precision floating point.

Definition at line 69 of file mintyp.h.

Data Structure Documentation

5 File Documentation

5.1 minimgio.h File Reference

MinImgIO library application programming interface.

Data Structures

- struct ExtImgProps

  Specifies additional information for an image. More...

Defines

- #define IS_BY_DEFAULT(a)

  Specifies a default value for a parameter of a function.
- #define MINIMGIO_API

  Specifies storage-class information (only for MSC).

Enumerations

- enum ImgFileFormat

  Specifies supported file formats.
- enum ImgFileComp

  Specifies supported TIFF compressions.

Functions

- int GuessImageFileFormat (const char *pFileName)

  Detect the image file format by the magic bytes or by the name.
- int GetMinImageFilePages (const char *pFileName)

  Returns the number of pages.
- int GetMinImageFileProps (MinImg *pImg, const char *pFileName, int page)

  Gets basic information about an image.
- int GetMinImageFilePropsEx (MinImg *pImg, ExtImgProps *pProps, const char *pFileName, int page)

  Gets detailed information about an image.
- int LoadMinImage (const MinImg *pImg, const char *pFileName, int page)

  Loads an image from a file.
- int SaveMinImage (const char *pFileName, const MinImg *pImg, int page)

  Saves an image to a specified file.

- int SaveMinImageEx (const char *pFileName, const MinImg *pImg, const ExtImgProps *pProps, int page)

  Saves an image to a specified file with specified options.
- int PackMinImage (const MinImg *pDst, const MinImg *pSrc, uint8_t level)

  Packs a grayscale image into monochrome once.
- int UnpackMinImage (const MinImg *pDst, const MinImg *pSrc)

  Unpacks a monochrome image into grayscale one.

5.1.1 Detailed Description

Definition in file minimgio.h.

5.2 minimgio.h

```
00001 /*
00002
00003 Copyright (c) 2011, Smart Engines Limited. All rights reserved.
00004
00005 All rights reserved.
00006
00007 Redistribution and use in source and binary forms, with or without
       modification,
00008 are permitted provided that the following conditions are met:
00009
00010    1. Redistributions of source code must retain the above copyright notice,
00011       this list of conditions and the following disclaimer.
00012
00013    2. Redistributions in binary form must reproduce the above copyright notice,
00014       this list of conditions and the following disclaimer in the documentation
00015       and/or other materials provided with the distribution.
00016
00017 THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESS OR
00018 IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
00019 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
       EVENT
00020 SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
00021 INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
00022 LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
00023 PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
00024 LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
00025 OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
00026 ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
00027
00028 The views and conclusions contained in the software and documentation are those
00029 of the authors and should not be interpreted as representing official policies,
00030 either expressed or implied, of copyright holders.
00031
00032 */
00033
00155 #pragma once
00156
00157 #ifndef MINIMGIO_H_INCLUDED
00158 #define MINIMGIO_H_INCLUDED
00159
00166 #ifdef IS_BY_DEFAULT
00167 #  undef IS_BY_DEFAULT
00168 #endif
00169
00170 #ifdef __cplusplus
00171 #  define IS_BY_DEFAULT(a) = a
00172   extern "C" {
00173 #else
00174 #  define IS_BY_DEFAULT(a)
00175 #endif
```

```
00176
00177 #include <minutils/minimg.h>
00178
00185 #ifdef _MSC_VER
00186 #  ifdef MINIMGIO_EXPORTS
00187 #    define MINIMGIO_API __declspec(dllexport)
00188 #  else
00189 #    define MINIMGIO_API __declspec(dllimport)
00190 #  endif
00191 #else
00192 #  define MINIMGIO_API
00193 #endif
00194
00208 typedef enum
00209 {
00210   IFF_UNKNOWN,
00211   IFF_TIFF,
00212   IFF_JPEG
00213 } ImgFileFormat;
00214
00220 typedef enum
00221 {
00222   IFC_NONE,
00223   IFC_LZW,
00224   IFC_GROUP4
00225 } ImgFileComp;
00226
00236 typedef struct
00237 {
00238   ImgFileFormat  iff;
00239   ImgFileComp    comp;
00240   float          xDPI;
00241   float          yDPI;
00242   int            qty;
00243 } ExtImgProps;
00244
00259 MINIMGIO_API int GuessImageFileFormat
00260 (
00261   const char *pFileName
00262 );
00263
00272 MINIMGIO_API int GetMinImageFilePages
00273 (
00274   const char *pFileName
00275 );
00276
00290 MINIMGIO_API int GetMinImageFileProps
00291 (
00292   MinImg     *pImg,
00293   const char *pFileName,
00294   int         page IS_BY_DEFAULT(0)
00295 );
00296
00311 MINIMGIO_API int GetMinImageFilePropsEx
00312 (
00313   MinImg      *pImg,
00314   ExtImgProps *pProps,
00315   const char  *pFileName,
00316   int          page IS_BY_DEFAULT(0)
00317 );
00318
00331 MINIMGIO_API int LoadMinImage
00332 (
00333   const MinImg *pImg,
00334   const char   *pFileName,
00335   int           page IS_BY_DEFAULT(0)
00336 );
00337
00349 MINIMGIO_API int SaveMinImage
00350 (
00351   const char   *pFileName,
00352   const MinImg *pImg,
00353   int           page IS_BY_DEFAULT(0)
00354 );
```

```
00355
00371 MINIMGIO_API int SaveMinImageEx
00372 (
00373   const char        *pFileName,
00374   const MinImg      *pImg,
00375   const ExtImgProps *pProps,
00376   int                page IS_BY_DEFAULT(0)
00377 );
00378
00391 MINIMGIO_API int PackMinImage
00392 (
00393   const MinImg *pDst,
00394   const MinImg *pSrc,
00395   uint8_t       level IS_BY_DEFAULT(128)
00396 );
00397
00408 MINIMGIO_API int UnpackMinImage
00409 (
00410   const MinImg *pDst,
00411   const MinImg *pSrc
00412 );
00413
00414
00415 #ifdef __cplusplus
00416   } // extern "C"
00417 #  undef IS_BY_DEFAULT
00418 #endif
00419
00420 #endif  /* MINIMGIO_H_INCLUDED */
```

5.3 minutils/minerr.h File Reference

Definition of possible return values.

Defines

- #define PROPAGATE_ERROR(call)

  If function failed then propagate the error code.
- #define SHOULD_WORK(call)

  If function failed then propagate INTERNAL_ERROR.

Enumerations

- enum MinErr

  Specifies basic error codes.

5.3.1 Detailed Description

Definition in file minerr.h.

5.4 minutils/minerr.h

```
00001 /*
00002
00003 Copyright (c) 2011, Smart Engines Limited. All rights reserved.
```

```
00004
00005 All rights reserved.
00006
00007 Redistribution and use in source and binary forms, with or without
       modification,
00008 are permitted provided that the following conditions are met:
00009
00010    1. Redistributions of source code must retain the above copyright notice,
00011       this list of conditions and the following disclaimer.
00012
00013    2. Redistributions in binary form must reproduce the above copyright notice,
00014       this list of conditions and the following disclaimer in the documentation
00015       and/or other materials provided with the distribution.
00016
00017 THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS ``AS IS'' AND ANY EXPRESS OR
00018 IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
00019 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
       EVENT
00020 SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
00021 INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
00022 LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
00023 PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
00024 LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
00025 OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
00026 ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
00027
00028 The views and conclusions contained in the software and documentation are those
00029 of the authors and should not be interpreted as representing official policies,
00030 either expressed or implied, of copyright holders.
00031
00032 */
00033
00039 #pragma once
00040
00041 #ifndef MINERR_H_INCLUDED
00042 #define MINERR_H_INCLUDED
00043
00062 typedef enum
00063 {
00064   NO_ERRORS       =  0,
00065
00066   BAD_ARGS        = -1,
00067
00068   NO_MEMORY       = -2,
00069
00070   NOT_IMPLEMENTED = -3,
00071
00072   INTERNAL_ERROR  = -4,
00073
00074   FILE_ERROR      = -5
00075
00076
00077 } MinErr;
00078
00085 #define PROPAGATE_ERROR(call) \
00086 { \
00087   int res = call; \
00088   if (res < 0) \
00089     return res; \
00090 }
00091
00098 #define SHOULD_WORK(call) \
00099 { \
00100   int res = call; \
00101   if (res < 0) \
00102     return INTERNAL_ERROR; \
00103 }
00104
00105 #endif  /* MINERR_H_INCLUDED */
```

5.5 minutils/minimg.h File Reference

Definition of a low-level representation of a bitmap image.

Data Structures

- struct MinImg

  A low-level universal representation of a bitmap image. More...

5.5.1 Detailed Description

Definition in file minimg.h.

5.6 minutils/minimg.h

```
00001 /*
00002
00003 Copyright (c) 2011, Smart Engines Limited. All rights reserved.
00004
00005 All rights reserved.
00006
00007 Redistribution and use in source and binary forms, with or without
       modification,
00008 are permitted provided that the following conditions are met:
00009
00010    1. Redistributions of source code must retain the above copyright notice,
00011       this list of conditions and the following disclaimer.
00012
00013    2. Redistributions in binary form must reproduce the above copyright notice,
00014       this list of conditions and the following disclaimer in the documentation
00015       and/or other materials provided with the distribution.
00016
00017 THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESS OR
00018 IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
00019 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
       EVENT
00020 SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
00021 INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
00022 LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
00023 PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
00024 LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
00025 OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
00026 ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
00027
00028 The views and conclusions contained in the software and documentation are those
00029 of the authors and should not be interpreted as representing official policies,
00030 either expressed or implied, of copyright holders.
00031
00032 */
00033
00039 #pragma once
00040
00041 #ifndef MINIMG_H_INCLUDED
00042 #define MINIMG_H_INCLUDED
00043
00044 #include <minutils/mintyp.h>
00045
00069 typedef struct
00070 {
00071   int32_t  width;
00072   int32_t  height;
00073   int32_t  stride;
00074   int32_t  channels;
```

```
00075   int32_t  channelDepth;
00076   MinFmt   format;
00077   uint8_t *pScan0;
00078 } MinImg;
00079
00080 #endif  /* MINIMG_H_INCLUDED */
```

5.7 minutils/mintyp.h File Reference

Definition of acceptable image types.

Data Structures

- struct float16

  Specifies half-precision floating point. More...

Typedefs

- typedef float16 float16_t

  Specifies `float16` *as* `float16_t`.
- typedef float float32_t

  Specifies `float` *as* `float32_t` *type.*
- typedef double float64_t

  Specifies `double` *as* `float64_t` *type.*

Enumerations

- enum MinFmt

  Specifies acceptable element formats of each individual channel.
- enum MinTyp

  Specifies acceptable element types of each individual channel.

5.7.1 Detailed Description

Definition in file mintyp.h.

5.8 minutils/mintyp.h

```
00001 /*
00002
00003 Copyright (c) 2011, Smart Engines Limited. All rights reserved.
00004
00005 All rights reserved.
00006
00007 Redistribution and use in source and binary forms, with or without
       modification,
00008 are permitted provided that the following conditions are met:
00009
00010    1. Redistributions of source code must retain the above copyright notice,
```

```
00011       this list of conditions and the following disclaimer.
00012
00013    2. Redistributions in binary form must reproduce the above copyright notice,
00014       this list of conditions and the following disclaimer in the documentation
00015       and/or other materials provided with the distribution.
00016
00017 THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS ``AS IS'' AND ANY EXPRESS OR
00018 IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
00019 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
      EVENT
00020 SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
00021 INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
00022 LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
00023 PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
00024 LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
00025 OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
00026 ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
00027
00028 The views and conclusions contained in the software and documentation are those
00029 of the authors and should not be interpreted as representing official policies,
00030 either expressed or implied, of copyright holders.
00031
00032 */
00033
00039 #pragma once
00040
00041 #ifndef MINTYP_H_INCLUDED
00042 #define MINTYP_H_INCLUDED
00043
00044 #ifdef _MSC_VER
00045 #include <minutils/stdint-vc.h>
00046 #else
00047 #include <stdint.h>
00048 #endif // _MSC_VER
00049
00056 typedef enum
00057 {
00058   FMT_UINT,
00059   FMT_INT,
00060   FMT_FLOAT
00061 } MinFmt;
00062
00069 typedef enum
00070 {
00071   TYP_UINT1,
00072   TYP_UINT8,
00073   TYP_INT8,
00074   TYP_UINT16,
00075   TYP_INT16,
00076   TYP_FLOAT16,
00077   TYP_UINT32,
00078   TYP_INT32,
00079   TYP_FLOAT32,
00080   TYP_UINT64,
00081   TYP_INT64,
00082   TYP_FLOAT64
00083 } MinTyp;
00084
00090 typedef struct
00091 {
00092   uint16_t significand : 10;
00093   uint16_t exponent    : 5;
00094   uint16_t sign        : 1;
00095 } float16;
00096
00097 typedef float16 float16_t;
00098 typedef float   float32_t;
00099 typedef double  float64_t;
00100
00101 #endif  /* MINTYP_H_INCLUDED */
```