

sclmageHandler.dll

構築: Doxygen 1.8.6

2014 年 04 月 15 日 (火) 21 時 56 分 52 秒



# Contents

<b>1</b>	<b>総合概要</b>	<b>1</b>
<b>2</b>	<b>名前空間索引</b>	<b>3</b>
2.1	名前空間一覧	3
<b>3</b>	<b>クラス索引</b>	<b>5</b>
3.1	クラス一覧	5
<b>4</b>	<b>名前空間詳解</b>	<b>7</b>
4.1	ns_scMetaImg 名前空間	7
4.1.1	詳解	7
4.1.2	列挙型詳解	7
4.1.2.1	anonymous enum	7
4.2	sclmageHandler 名前空間	7
4.2.1	詳解	8
4.2.2	関数詳解	8
4.2.2.1	sclmageHandler_Create	8
4.2.2.2	sclmageHandler_Delete	8
4.2.2.3	sclmageHandler_LastErrorMessage	8
4.2.2.4	sclmageHandler_Save	8
4.2.2.5	sclmageHandler_SetQuality	9
4.3	sclmageHandler::FORMAT 名前空間	9
4.3.1	詳解	9
4.3.2	列挙型詳解	9
4.3.2.1	anonymous enum	9
<b>5</b>	<b>クラス詳解</b>	<b>11</b>
5.1	_MY_NULL_CLASS_ クラス	11
5.1.1	詳解	11
5.1.2	関数詳解	11
5.1.2.1	operator T *	11
5.1.2.2	operator T C::*	11
5.1.2.3	operator&	11

5.2	SC_METAIMG_PALETTE 構造体	12
5.2.1	詳解	12
5.3	scMetalmg クラス	12
5.3.1	詳解	13
5.3.2	構築子と解体子	13
5.3.2.1	scMetalmg	13
5.3.2.2	scMetalmg	13
5.3.3	関数詳解	13
5.3.3.1	Bit	13
5.3.3.2	Channel	14
5.3.3.3	CstImgPtr08	14
5.3.3.4	CstImgPtr08_2d	14
5.3.3.5	CstImgPtr16	14
5.3.3.6	CstImgPtr16_2d	14
5.3.3.7	CstPalette	14
5.3.3.8	Height	14
5.3.3.9	ImgPtr	14
5.3.3.10	ImgPtr08	15
5.3.3.11	ImgPtr08_2d	15
5.3.3.12	ImgPtr16	15
5.3.3.13	ImgPtr16_2d	15
5.3.3.14	MemCpy__	15
5.3.3.15	operator=	15
5.3.3.16	Palette	15
5.3.3.17	PaletteSize	16
5.3.3.18	Resize	16
5.3.3.19	Resize	16
5.3.3.20	Width	16
5.3.3.21	~ scMetalmg	16
5.3.4	メンバ詳解	16
5.3.4.1	ary_palette_	16
5.3.4.2	bit_	17
5.3.4.3	buf1d_	17
5.3.4.4	buf2d_	17
5.3.4.5	channel_	17
5.3.4.6	height_	17
5.3.4.7	palette_size_	17
5.3.4.8	width_	17

# Chapter 1

## 総合概要

doxygen で生成した、sclmageHandler.dll 資料のメインページです。



## Chapter 2

# 名前空間索引

### 2.1 名前空間一覧

詳解が付いた名前空間の一覧です。

<b>ns_scMetalmg</b>		
定数を定義	.....	7
<b>sclmageHandler</b>		
画像ファイルを簡単に読み書きするための関数群	.....	7
<b>sclmageHandler::FORMAT</b>		
対応画像フォーマット	.....	9



## Chapter 3

# クラス索引

### 3.1 クラス一覧

クラス・構造体・共用体・インターフェースの一覧です。

<b>_MY_NULL_CLASS_</b>		
ユーザ定義 NULL	.....	11
<b>SC_METAIMG_PALETTE</b>		
パレットを扱うための構造体	.....	12
<b>scMetaimg</b>		
抽象画像データを扱うためのクラス	.....	12



## Chapter 4

# 名前空間詳解

### 4.1 ns\_scMetalmg 名前空間

定数を定義

列挙型

- enum { **RED** = 0, **GREEN** = 1, **BLUE** = 2 }

#### 4.1.1 詳解

定数を定義 この名前空間は、scMetalmg クラスで扱う定数を定義する。

#### 4.1.2 列挙型詳解

##### 4.1.2.1 anonymous enum

列挙値

**RED** Red Channel = 0.

**GREEN** Green Channel = 1.

**BLUE** Blue Channel = 2.

### 4.2 sclmageHandler 名前空間

画像ファイルを簡単に読み書きするための関数群

名前空間

- **FORMAT**  
対応画像フォーマット

関数

- const char \*\_\_stdcall **sclmageHandler\_LastErrorMessage** ()
- int \_\_stdcall **sclmageHandler\_SetQuality** (int quality)

- int \_\_stdcall **sclmageHandler\_Save** (const char \*img\_filename, const **scMetalmg** \*p\_meta\_img, int img\_format\_id)
- void \* \_\_stdcall **sclmageHandler\_Create** (const char \*img\_filename)
- int \_\_stdcall **sclmageHandler\_Delete** (**scMetalmg** \*p\_meta\_img)

#### 4.2.1 詳解

画像ファイルを簡単に読み書きするための関数群 この関数群は、画像ファイルを簡単に読み書きするための関数、定数をまとめたものである。この関数群は、sclmageHandler.dll によって提供される。

#### 4.2.2 関数詳解

##### 4.2.2.1 void\* \_\_stdcall sclmageHandler::sclmageHandler\_Create ( const char \* img\_filename )

この関数は、画像ファイルを読み込み、scMetalmg を構築する。

引数

in	<i>img_filename</i>	画像ファイルのパス
----	---------------------	-----------

戻り値

成功した場合、ヒープ上に構築した **scMetalmg** (p. 12) のアドレスを void\* 形式で帰す。失敗した場合、NULL を帰す。

##### 4.2.2.2 int \_\_stdcall sclmageHandler::sclmageHandler\_Delete ( scMetalmg \* p\_meta\_img )

この関数は、scMetalmg \* を破棄する。

引数

in	<i>p_meta_img</i>	破棄される画像データ
----	-------------------	------------

戻り値

無し

##### 4.2.2.3 const char\* \_\_stdcall sclmageHandler::sclmageHandler\_LastErrorMessage ( )

この関数は、最後に発生したエラーメッセージを帰す。

戻り値

最後に発生したエラーメッセージを帰す。最後に実行した関数が正常終了した場合、正常終了を示すメッセージを帰す。

##### 4.2.2.4 int \_\_stdcall sclmageHandler::sclmageHandler\_Save ( const char \* img\_filename, const scMetalmg \* p\_meta\_img, int img\_format\_id )

この関数は、画像フォーマットを指定して、画像ファイルを保存する。現時点で、この関数は以下の組み合わせに対応している。

(channel,bit) = ( 1, 1 ), ( 1, 8 ), ( 3, 8 )

```
typedef int ( __stdcall *P_SAVE_IMAGE )( const char *, const scMetaImg *, int );
P_SAVE_IMAGE p_save_img;
p_save_img = reinterpret_cast< P_SAVE_IMAGE >( ::GetProcAddress( hDLL, "scImageHandler_Save" ) );
if( p_save_img != NULL )
{
    // scMetaImg test_img は初期化済み
    ( *p_save_img )( filename_img, &test_img, scImageHandler::FORMAT::JPEG );
}

```

引数

in	<i>img_filename</i>	画像ファイルのパス
in	<i>p_meta_img</i>	保存される画像データ
in	<i>img_format_id</i>	画像フォーマット

参照

**FORMAT::JPEG** (p. 9)  
**FORMAT::PNG** (p. 10)  
**FORMAT::GIF** (p. 10)  
**FORMAT::TIFF** (p. 10)  
**FORMAT::BMP** (p. 10)

戻り値

成功した場合、0 を帰す。失敗した場合、0 以外 を帰す。

#### 4.2.25 int \_\_stdcall scImageHandler::scImageHandler\_SetQuality ( int quality )

この関数は、JPEG 形式で保存される画像の画質指定パラメータ 0(最低)~ 100(最高) の整数値を指定する。初期値は 75。

戻り値

変更前の画質指定パラメータを帰す。

## 4.3 scImageHandler::FORMAT 名前空間

対応画像フォーマット

列挙型

```
enum {
    JPEG = 0, PNG, TIFF, GIF,
    BMP, _IMG_FORMAT_RESERVED_ }

```

### 4.3.1 詳解

対応画像フォーマット scImageHandler.dll が対応する画像フォーマットを示す列挙型

### 4.3.2 列挙型詳解

#### 4.3.2.1 anonymous enum

列挙値

**JPEG** Jpeg.

**PNG** Png.

**TIFF** Tiff.

**GIF** Gif.

**BMP** Windows Bitmap.

**\_IMG\_FORMAT\_RESERVED\_** 未使用

## Chapter 5

# クラス詳解

### 5.1 `_MY_NULL_CLASS_` クラス

ユーザ定義 NULL

```
#include <my_null.h>
```

公開メンバ関数

- `template<class T >`  
**`operator T * ( ) const`**
- `template<class C , class T >`  
**`operator T C::* ( ) const`**

非公開メンバ関数

- `void operator& ( ) const`

#### 5.1.1 詳解

ユーザ定義 NULL

C++ の 0 と NULL の混同を避けるためのクラス。

Effective C++ 25 章参照

#### 5.1.2 関数詳解

**5.1.2.1 `template<class T > _MY_NULL_CLASS_::operator T * ( ) const` [inline]**

あらゆる型の非メンバへのポインタに変換する。

**5.1.2.2 `template<class C , class T > _MY_NULL_CLASS_::operator T C::* ( ) const` [inline]**

あらゆる型のメンバへのポインタに変換する。

**5.1.2.3 `void _MY_NULL_CLASS_::operator& ( ) const` [private]**

private 関数にすることで、アドレス取得を不可能にする。

このクラス詳解は次のファイルから抽出されました:

- my\_null.h

## 5.2 SC\_METAIMG\_PALETTE 構造体

パレットを扱うための構造体

```
#include <scMetaImg.h>
```

公開変数類

- unsigned char **Red**  
パレット情報の赤成分
- unsigned char **Green**  
パレット情報の緑成分
- unsigned char **Blue**  
パレット情報の青成分
- unsigned char **Reserved**  
パレット情報の未使用領域

### 5.2.1 詳解

パレットを扱うための構造体

この構造体は、scMetaImg クラスのパレットを扱うための機能を提供する。

この構造体詳解は次のファイルから抽出されました:

- scMetaImg.h

## 5.3 scMetaImg クラス

抽象画像データを扱うためのクラス

```
#include <scMetaImg.h>
```

公開メンバ関数

- **scMetaImg** ()
- virtual **~scMetaImg** ()
- **scMetaImg & operator=** (const **scMetaImg** &rhs)
- int **Width** () const
- int **Height** () const
- int **Bit** () const
- int **Channel** () const
- **SC\_METAIMG\_PALETTE \* Palette** ()
- const **SC\_METAIMG\_PALETTE \* CstPalette** () const
- int **PaletteSize** () const
- int **Resize** (int width, int height, int bit, int channel)
- int **Resize** (const **scMetaImg** \*ref\_img)
- void \* **ImgPtr** (int index) const

- unsigned char \* **ImgPtr08** (int index) const
- unsigned short \* **ImgPtr16** (int index) const
- const unsigned char \* **CstImgPtr08** (int index) const
- const unsigned short \* **CstImgPtr16** (int index) const
- unsigned char \*\* **ImgPtr08\_2d** () const
- unsigned short \*\* **ImgPtr16\_2d** () const
- const unsigned char \*\* **CstImgPtr08\_2d** () const
- const unsigned short \*\* **CstImgPtr16\_2d** () const

#### 限定公開メンバ関数

- **scMetalmg** (const **scMetalmg** &rhs)
- void \* **MemCpy\_\_** (void \*buf1, const void \*buf2, int byte\_size)

#### 限定公開変数類

- void \* **buf1d\_**
- void \*\* **buf2d\_**
- int **width\_**
- int **height\_**
- int **bit\_**
- int **channel\_**
- **SC\_METAIMG\_PALETTE** \* **ary\_palette\_**
- int **palette\_size\_**

### 5.3.1 詳解

抽象画像データを扱うためのクラス

このクラスは、抽象画像データを扱うための機能を提供する。画素データは void 型配列によって格納される。画素データが格納されるメモリは、コンストラクタ、代入、**Resize** 関数によって自動的に確保される。代入、**Resize** 関数によって元の配列は破棄され、画素格納アドレスが変更されることに留意すること。

### 5.3.2 構築子と解体子

5.3.2.1 **scMetalmg::scMetalmg ( const **scMetalmg** & rhs )** [protected]

コピーコンストラクタ

5.3.2.2 **scMetalmg::scMetalmg ( )** [inline]

コンストラクタ

### 5.3.3 関数詳解

5.3.3.1 **int scMetalmg::Bit ( ) const** [inline]

戻り値

単色のビット数を帰す。

### 5.3.3.2 int scMetalmg::Channel ( ) const [inline]

戻り値

色数を帰す。

### 5.3.3.3 const unsigned char\* scMetalmg::CstlmgPtr08 ( int *index* ) const [inline]

引数

in	<i>index</i>	色の指定
----	--------------	------

戻り値

指定した色の画素データの先頭アドレスを const unsigned char \* で帰す。

### 5.3.3.4 const unsigned char\*\* scMetalmg::CstlmgPtr08\_2d ( ) const [inline]

戻り値

各色の画素データの先頭アドレスを const unsigned char \*\* で帰す。

### 5.3.3.5 const unsigned short\* scMetalmg::CstlmgPtr16 ( int *index* ) const [inline]

引数

in	<i>index</i>	色の指定
----	--------------	------

戻り値

指定した色の画素データの先頭アドレスを const unsigned short \* で帰す。

### 5.3.3.6 const unsigned short\*\* scMetalmg::CstlmgPtr16\_2d ( ) const [inline]

戻り値

各色の画素データの先頭アドレスを const unsigned short \*\* で帰す。

### 5.3.3.7 const SC\_METAIMG\_PALETTE\* scMetalmg::CstPalette ( ) const [inline]

戻り値

パレットのアドレスを const **SC\_METAIMG\_PALETTE** (p.12) \* で帰す。

### 5.3.3.8 int scMetalmg::Height ( ) const [inline]

戻り値

画像高さを帰す。

### 5.3.3.9 void\* scMetalmg::lmgPtr ( int *index* ) const [inline]

引数

in	<i>index</i>	色の指定
----	--------------	------

戻り値

指定した色の画素データの先頭アドレスを void \* で帰す。

#### 5.3.3.10 unsigned char\* scMetalmg::lmgPtr08 ( int *index* ) const [inline]

引数

in	<i>index</i>	色の指定
----	--------------	------

戻り値

指定した色の画素データの先頭アドレスを unsigned char \* で帰す。

#### 5.3.3.11 unsigned char\*\* scMetalmg::lmgPtr08\_2d ( ) const [inline]

戻り値

各色の画素データの先頭アドレスを unsigned char \*\* で帰す。

#### 5.3.3.12 unsigned short\* scMetalmg::lmgPtr16 ( int *index* ) const [inline]

引数

in	<i>index</i>	色の指定
----	--------------	------

戻り値

指定した色の画素データの先頭アドレスを unsigned short \* で帰す。

#### 5.3.3.13 unsigned short\*\* scMetalmg::lmgPtr16\_2d ( ) const [inline]

各色の画素データの先頭アドレスを unsigned short \*\* で帰す。

#### 5.3.3.14 void\* scMetalmg::MemCpy\_\_ ( void \* *buf1*, const void \* *buf2*, int *byte\_size* ) [inline],[protected]

メモリコピー

#### 5.3.3.15 scMetalmg& scMetalmg::operator= ( const scMetalmg & *rhs* ) [inline]

代入演算子 自分自身への代入を指定された場合、何もしない。

#### 5.3.3.16 SC\_METAIMG\_PALETTE\* scMetalmg::Palette ( ) [inline]

戻り値

パレットのアドレスを帰す。

### 5.3.3.17 int scMetalmg::PaletteSize ( ) const [inline]

戻り値

パレットサイズ (個数) を帰す。

### 5.3.3.18 int scMetalmg::Resize ( int width, int height, int bit, int channel ) [inline]

画像情報の更新とメモリの確保を行なう。関数呼び出し前の画素データは破棄される。メモリサイズ ( buf\_size、バイト数 ) は、以下のように算出される。

```
pix_size = width * height;
plane_size = pix_size * ( ( bit + 7 ) >> 3 );
buf_size = plane_size * channel;
```

引数

in	<i>width</i>	新しい画像の幅
in	<i>height</i>	新しい画像の高さ
in	<i>bit</i>	新しい画像の単色のビット数
in	<i>channel</i>	新しい画像の色数

戻り値

成功した場合、0 を帰す。失敗した場合、0 以外を帰す。

### 5.3.3.19 int scMetalmg::Resize ( const scMetalmg \* ref\_img ) [inline]

入力した scMetalmg 型変数を参照し、画像情報と更新とメモリの確保を行なう。画像情報とメモリサイズは、参照するインスタンスと同じになる。関数呼び出し前の画素データは破棄される。

引数

in	<i>ref_img</i>	参照する scMetalmg 型変数のポインタ
----	----------------	-------------------------

戻り値

成功した場合、0 を帰す。失敗した場合、0 以外を帰す。

### 5.3.3.20 int scMetalmg::Width ( ) const [inline]

戻り値

画像幅を帰す。

### 5.3.3.21 virtual scMetalmg::~scMetalmg ( ) [inline],[virtual]

デストラクタ buf1d\_、buf2d\_ が NULL では無い場合、これらを解放する。

## 5.3.4 メンバ詳解

### 5.3.4.1 SC\_METAIMG\_PALETTE\* scMetalmg::ary\_palette\_ [protected]

パレット

#### 5.3.4.2 int scMetalmg::bit\_ [protected]

単色 1 画素の情報量

現状では、1bit、8 bit、16 bit のみサポートしている。

#### 5.3.4.3 void\* scMetalmg::buf1d\_ [protected]

画素データを格納するバッファの先頭アドレス。各色の画素データが面順次でシーケンシャルに格納される。画像原点は左上。

ラインデータのサイズは、1 バイトの倍数でなければならない。  $line\_bytesize = (width * bit + 7) / 8$  width : 横方向の画素数 bit : 単色 1 画素の情報量

#### 5.3.4.4 void\*\* scMetalmg::buf2d\_ [protected]

画素データの各色の先頭アドレス

#### 5.3.4.5 int scMetalmg::channel\_ [protected]

色数 RGB 画像なら色数=3、インデックスカラーなら色数=1

#### 5.3.4.6 int scMetalmg::height\_ [protected]

画像の高さ

#### 5.3.4.7 int scMetalmg::palette\_size\_ [protected]

パレットサイズ

#### 5.3.4.8 int scMetalmg::width\_ [protected]

画像の幅

このクラス詳解は次のファイルから抽出されました:

- scMetalmg.h

# Index

- `_IMG_FORMAT_RESERVED_`
    - `scImageHandler::FORMAT`, 10
  - `_MY_NULL_CLASS_`, 11
    - `operator T *`, 11
    - `operator T C::*`, 11
    - `operator&`, 11
- `ary_palette_`
  - `scMetalmg`, 16
- BLUE
  - `ns_scMetalmg`, 7
- BMP
  - `scImageHandler::FORMAT`, 10
- Bit
  - `scMetalmg`, 13
- `bit_`
  - `scMetalmg`, 16
- `buf1d_`
  - `scMetalmg`, 17
- `buf2d_`
  - `scMetalmg`, 17
- Channel
  - `scMetalmg`, 13
- `channel_`
  - `scMetalmg`, 17
- `CstImgPtr08`
  - `scMetalmg`, 14
- `CstImgPtr08_2d`
  - `scMetalmg`, 14
- `CstImgPtr16`
  - `scMetalmg`, 14
- `CstImgPtr16_2d`
  - `scMetalmg`, 14
- `CstPalette`
  - `scMetalmg`, 14

- GIF
- `scImageHandler::FORMAT`, 10
- GREEN
- `ns_scMetalmg`, 7
- Height
- `scMetalmg`, 14
- `height_`
- `scMetalmg`, 17
- ImgPtr
- `scMetalmg`, 14
- ImgPtr08
- `scMetalmg`, 15
- ImgPtr08\_2d
- `scMetalmg`, 15
- ImgPtr16
- `scMetalmg`, 15
- ImgPtr16\_2d
- `scMetalmg`, 15
- JPEG
- `scImageHandler::FORMAT`, 9
- MemCpy\_
- `scMetalmg`, 15
- `ns_scMetalmg`, 7
- BLUE, 7
- GREEN, 7
- RED, 7
- `operator T *`
- `_MY_NULL_CLASS_`, 11
- `operator T C::*`
- `_MY_NULL_CLASS_`, 11
- `operator=`
- `scMetalmg`, 15
- `operator&`
- `_MY_NULL_CLASS_`, 11
- PNG
- `scImageHandler::FORMAT`, 9
- Palette
- `scMetalmg`, 15
- `palette_size_`
- `scMetalmg`, 17
- PaletteSize
- `scMetalmg`, 15
- RED
- `ns_scMetalmg`, 7
- Resize
- `scMetalmg`, 16
- SC\_METAIMG\_PALETTE, 12
- `scImageHandler`, 7
  - `scImageHandler_Create`, 8
  - `scImageHandler_Delete`, 8
  - `scImageHandler_LastErrorMessage`, 8
  - `scImageHandler_Save`, 8
  - `scImageHandler_SetQuality`, 9
- `scImageHandler::FORMAT`, 9
- `_IMG_FORMAT_RESERVED_`, 10

- BMP, 10
- GIF, 10
- JPEG, 9
- PNG, 9
- TIFF, 10
- scImageHandler\_Create
  - scImageHandler, 8
- scImageHandler\_Delete
  - scImageHandler, 8
- scImageHandler\_LastErrorMessage
  - scImageHandler, 8
- scImageHandler\_Save
  - scImageHandler, 8
- scImageHandler\_SetQuality
  - scImageHandler, 9
- scMetaImg, 12
  - ary\_palette\_, 16
  - Bit, 13
  - bit\_, 16
  - buf1d\_, 17
  - buf2d\_, 17
  - Channel, 13
  - channel\_, 17
  - CstImgPtr08, 14
  - CstImgPtr08\_2d, 14
  - CstImgPtr16, 14
  - CstImgPtr16\_2d, 14
  - CstPalette, 14
  - Height, 14
  - height\_, 17
  - ImgPtr, 14
  - ImgPtr08, 15
  - ImgPtr08\_2d, 15
  - ImgPtr16, 15
  - ImgPtr16\_2d, 15
  - MemCpy\_, 15
  - operator=, 15
  - Palette, 15
  - palette\_size\_, 17
  - PaletteSize, 15
  - Resize, 16
  - scMetaImg, 13
  - scMetaImg, 13
  - Width, 16
  - width\_, 17
  - scMetaImg, 16
- TIFF
  - scImageHandler::FORMAT, 10
- Width
  - scMetaImg, 16
- width\_
  - scMetaImg, 17
- scMetaImg
  - scMetaImg, 16