SDL_image

12 June 2003

Jonathan Atkins

The latest copy of this document can be found at `http://jcatki.no-ip.org/SDL_image`

Table of Contents

1 Overview

A Little Bit About Me

I am currently, as I write this document, a programmer for Raytheon. There I do all sorts of communications, network, GUI, and other general programming tasks in C/C++ on the Solaris and sometimes Linux Operating Systems.

Feel free to contact me: jcatki@jcatki.no-ip.org

I am also usually on IRC at irc.freenode.net in the #SDL channel as LIM

Why is this for you?

Images provide the basic visual building blocks for any user interface. Colors and fun shapes are the stuff that we as kids looked at for hours at a time while trying to shoot down big aliens and rescue pixelated princesses. Now it's our turn to make the images that others will remember later in life perhaps. Now how do we get this dang images into our SDL programs, and be flexible in the handling of the images so that we don't even have to worry about what various formats they may be in? This is where SDLimage makes all of our lives easier. This document doesn't help you make artwork, but it will give you the functional knowledge on how to get that art into your game. Now go forth and make your Stick Figure of Justice, someone else might fill in for your lack of artistry, at least you won't have to make much of an effort to include the new and better art into your code.

This is the README in the SDL_image source archive.

```
SDL_image 1.2

The latest version of this library is available from:
SDL_image Homepage

This is a simple library to load images of various formats as SDL surfaces.
This library supports BMP, PNM (PPM/PGM/PBM), XPM, LBM, PCX, GIF, JPEG, PNG,
TGA, and TIFF formats.

API:
#include "SDL_image.h"

SDL_Surface *IMG_Load(const char *file);
or
SDL_Surface *IMG_Load_RW(SDL_RWops *src, int freesrc);
or
SDL_Surface *IMG_LoadTyped_RW(SDL_RWops *src, int freesrc, char *type);

where type is a string specifying the format (i.e. "PNG" or "pcx").
Note that IMG_Load_RW cannot load TGA images.

To create a surface from an XPM image included in C source, use:

SDL_Surface *IMG_ReadXPMFromArray(char **xpm);

An example program 'showimage' is included, with source in showimage.c

JPEG support requires the JPEG library:
    IJG Homepage
PNG support requires the PNG library:
    PNG Homepage
  and the Zlib library:
    Zlib Homepage
TIFF support requires the TIFF library:
    SGI TIFF FTP Site

This library is under the GNU Library General Public License, see the file
"COPYING" for details.  Certain image loaders may be under a different
license, see the individual image loader source files for details.
```

2 Getting Started

This assumes you have gotten SDL_image and installed it on your system. SDL_image has an README document in the source distribution to help you get it compiled and installed. Well it at least points you to locations for the source code of some of the image libraries SDL_image can use. Most of the other image formats are builtin to SDL_image.

Generally, in UNIX-like environments, installation consists of:

```
./configure
make
make install
```

SDL_image supports loading and decoding images from the following formats:

TGA TrueVision Targa (MUST have .tga)

BMP Windows Bitmap(.bmp)

PNM Portable Anymap (.pnm)
`.pbm` = Portable BitMap (mono)
`.pgm` = Portable GreyMap (256 greys)
`.ppm` = Portable PixMap (full color)

XPM X11 Pixmap (.xpm) can be #included directly in code
This is NOT the same as XBM(X11 Bitmap) format, which is for monocolor images.

XCF GIMP native (.xcf) (XCF = eXperimental Computing Facility?)
This format is always changing, and since there's no library supplied by the GIMP project to load XCF, the loader may frequently fail to load much of any image from an XCF file. It's better to load this in GIMP and convert to a better supported image format.

PCX ZSoft IBM PC Paintbrush (.pcx)

GIF CompuServe Graphics Interchange Format (.gif)

JPG Joint Photographic Experts Group JFIF format (.jpg or .jpeg)

TIF Tagged Image File Format (.tif or .tiff)

LBM Interleaved Bitmap (.lbm or .iff) FORM : ILBM or PBM(packed bitmap)
HAM6, HAM8, and 24bit types are not supported.

PNG Portable Network Graphics (.png)

You may also want to look at some demonstration code which may be downloaded from:
`http://jcatki.no-ip.org/SDL_image/`

2.1 Includes

To use SDL_image functions in a C/C++ source code file, you must use the SDL_image.h include file:

```
#include "SDL_image.h"
```

2.2 Compiling

To link with SDL_image you should use sdl-config to get the required SDL compilation options. After that, compiling with SDL_image is quite easy.
Note: Some systems may not have the SDL_image library and include file in the same place as the SDL library and includes are located, in that case you will need to add more -I and -L paths to these command lines.

Simple Example for compiling an object file:

```
cc -c ‘sdl-config --cflags‘ mysource.c
```

Simple Example for compiling an object file:

```
cc -o myprogram mysource.o ‘sdl-config --libs‘ -lSDL_image
```

Now `myprogram` is ready to run.

3 Functions

These are the functions in the SDL_image API.

3.1 Loading

These functions create an `SDL_Surface` from image data either from a file, or SDL_RWop, or from an array of data.

3.1.1 IMG_Load

`SDL_Surface *`**IMG_Load**`(const char *`*`file`*`)`

file Image file name to load a surface from.

Load *file* for use as an image in a new surface. This actually calls `IMG_LoadTyped_RW`, with the file extension used as the *type* string. This can load all supported image files, including TGA as long as the filename ends with ".tga". It is best to call this outside of event loops, and rather keep the loaded images around until you are really done with them, as disk speed and image conversion to a surface is not *that* speedy. Don't forget to SDL_FreeSurface the returned surface pointer when you are through with it.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors, such as no support built for the image, or a file reading error.

```
// load sample.png into image
SDL_Surface *image;
image=IMG_Load("sample.png");
if(!image) {
    printf("IMG_Load: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.2 [IMG_Load_RW], page 9, Section 3.1.3 [IMG_LoadTyped_RW], page 10

3.1.2 IMG_Load_RW

`SDL_Surface *`**IMG_Load_RW**`(SDL_RWops *`*src*`, int `*freesrc*`)`

src The source SDL_RWops as a pointer. The image is loaded from this.

freesrc A non-zero value mean is will automatically close/free the *src* for you.

Load *src* for use as a surface. This can load all supported image formats, **except TGA**. Using `SDL_RWops` is not covered here, but they enable you to load from almost any source.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors.

```
// load sample.png in to image
SDL_Surface *image;
image=IMG_Load_RW(SDL_RWFromFile("sample.png", "rb"), 1);
if(!image) {
    printf("IMG_Load_RW: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.1 [IMG_Load], page 8, Section 3.1.3 [IMG_LoadTyped_RW], page 10

3.1.3 IMG_LoadTyped_RW

`SDL_Surface` ***IMG_LoadTyped_RW**`(SDL_RWops *`*src*`, int` *freesrc*`, char *`*type*`)`

src The source SDL_RWops as a pointer. The image is loaded from this.

freesrc A non-zero value mean is will automatically close/free the *src* for you.

type A string that indicates which format type to interpret the image as.
Here is a list of the currently recognized strings (case is not important):

```
"TGA"
"BMP"
"PNM"
"XPM"
"XCF"
"PCX"
"GIF"
"JPG"
"TIF"
"LBM"
"PNG"
```

Load *src* for use as a surface. This can load all supported image formats. This method does not guarantee that the format specified by type is the format of the loaded image, except in the case when TGA format is specified (or any other non-magicable format). Using `SDL_RWops` is not covered here, but they enable you to load from almost any source.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors.

```
// load sample.tga into image
SDL_Surface *image;
image=IMG_Load_RW(SDL_RWFromFile("sample.tga", "rb"), 1, "TGA");
if(!image) {
    printf("IMG_Load_RW: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.1 [IMG_Load], page 8, Section 3.1.2 [IMG_Load_RW], page 9

3.1.4 IMG_LoadBMP_RW

`SDL_Surface *IMG_LoadBMP_RW(SDL_RWops *src)`

src The source SDL_RWops as a pointer. The BMP image is loaded from this.

Load *src* as a BMP image for use as a surface, if BMP support is compiled into the SDL_image library.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors, like if BMP is not supported, or a read error.

```
// load sample.bmp into image
SDL_Surface *image;
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.bmp", "rb");
image=IMG_LoadBMP_RW(rwop);
if(!image) {
    printf("IMG_LoadBMP_RW: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10, Section 3.2.1 [IMG_isBMP], page 24

3.1.5 IMG_LoadPNM_RW

`SDL_Surface` ***IMG_LoadPNM_RW**`(SDL_RWops *`*src*`)`

src The source SDL_RWops as a pointer. The PNM image is loaded from this.

Load *src* as a PNM image for use as a surface, if PNM support is compiled into the SDL_image library.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors, like if PNM is not supported, or a read error.

```
// load sample.pnm into image
SDL_Surface *image;
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.pnm", "rb");
image=IMG_LoadPNM_RW(rwop);
if(!image) {
    printf("IMG_LoadPNM_RW: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10, Section 3.2.2 [IMG_isPNM], page 25

3.1.6 IMG_LoadXPM_RW

`SDL_Surface *`**IMG_LoadXPM_RW**`(SDL_RWops *`*src*`)`

src The source SDL_RWops as a pointer. The XPM image is loaded from this.

Load *src* as a XPM image for use as a surface, if XPM support is compiled into the SDL_image library.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors, like if XPM is not supported, or a read error.

```
// load sample.xpm into image
SDL_Surface *image;
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.xpm", "rb");
image=IMG_LoadXPM_RW(rwop);
if(!image) {
    printf("IMG_LoadXPM_RW: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10, Section 3.1.15 [IMG_ReadXPMFromArray], page 22, Section 3.2.3 [IMG_isXPM], page 26

3.1.7 IMG_LoadXCF_RW

`SDL_Surface *`**IMG_LoadXCF_RW**`(SDL_RWops *`*src*`)`

src The source SDL_RWops as a pointer. The XCF image is loaded from this.

Load *src* as a XCF image for use as a surface, if XCF support is compiled into the SDL_image library.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors, like if XCF is not supported, or a read error.

```
// load sample.xcf into image
SDL_Surface *image;
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.xcf", "rb");
image=IMG_LoadXCF_RW(rwop);
if(!image) {
    printf("IMG_LoadXCF_RW: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10, Section 3.2.4 [IMG_isXCF], page 27

3.1.8 IMG_LoadPCX_RW

`SDL_Surface *`**IMG_LoadPCX_RW**`(SDL_RWops *`*src*`)`

src The source SDL_RWops as a pointer. The PCX image is loaded from this.

Load *src* as a PCX image for use as a surface, if PCX support is compiled into the SDL_image library.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors, like if PCX is not supported, or a read error.

```
// load sample.pcx into image
SDL_Surface *image;
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.pcx", "rb");
image=IMG_LoadPCX_RW(rwop);
if(!image) {
    printf("IMG_LoadPCX_RW: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10, Section 3.2.5 [IMG_isPCX], page 28

3.1.9 IMG_LoadGIF_RW

`SDL_Surface *`**IMG_LoadGIF_RW**`(SDL_RWops *`*src*`)`

src The source SDL_RWops as a pointer. The GIF image is loaded from this.

Load *src* as a GIF image for use as a surface, if GIF support is compiled into the SDL_image library.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors, like if GIF is not supported, or a read error.

```
// load sample.gif into image
SDL_Surface *image;
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.gif", "rb");
image=IMG_LoadGIF_RW(rwop);
if(!image) {
    printf("IMG_LoadGIF_RW: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10, Section 3.2.6 [IMG_isGIF], page 29

3.1.10 IMG_LoadJPG_RW

`SDL_Surface *`**IMG_LoadJPG_RW**`(SDL_RWops *`*src*`)`

src The source SDL_RWops as a pointer. The JPG image is loaded from this.

Load *src* as a JPG image for use as a surface, if JPG support is compiled into the SDL_image library.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors, like if JPG is not supported, or a read error.

```
// load sample.jpg into image
SDL_Surface *image;
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.jpg", "rb");
image=IMG_LoadJPG_RW(rwop);
if(!image) {
    printf("IMG_LoadJPG_RW: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10, Section 3.2.7 [IMG_isJPG], page 30

3.1.11 IMG_LoadTIF_RW

`SDL_Surface *`**IMG_LoadTIF_RW**`(SDL_RWops *`*src*`)`

src The source SDL_RWops as a pointer. The TIF image is loaded from this.

Load *src* as a TIF image for use as a surface, if TIF support is compiled into the SDL_image library.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors, like if TIF is not supported, or a read error.

```
// load sample.tif into image
SDL_Surface *image;
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.tif", "rb");
image=IMG_LoadTIF_RW(rwop);
if(!image) {
    printf("IMG_LoadTIF_RW: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10, Section 3.2.8 [IMG_isTIF], page 31

3.1.12 IMG_LoadPNG_RW

`SDL_Surface` ***IMG_LoadPNG_RW**(`SDL_RWops` **src*)

src The source SDL_RWops as a pointer. The PNG image is loaded from this.

Load *src* as a PNG image for use as a surface, if PNG support is compiled into the SDL_image library.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors, like if PNG is not supported, or a read error.

```
// load sample.png into image
SDL_Surface *image;
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.png", "rb");
image=IMG_LoadPNG_RW(rwop);
if(!image) {
    printf("IMG_LoadPNG_RW: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10, Section 3.2.9 [IMG_isPNG], page 32

3.1.13 IMG_LoadTGA_RW

`SDL_Surface *`**IMG_LoadTGA_RW**`(SDL_RWops *`*`src`*`)`

src The source SDL_RWops as a pointer. The TGA image is loaded from this.

Load *src* as a TGA image for use as a surface, if TGA support is compiled into the SDL_image library. If you try to load a non TGA image, you might succeed even when it's not TGA image formatted data, this is because the TGA has no magic, which is a way of identifying a filetype from a signature in it's contents. So be careful with this.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors, like if TGA is not supported, or a read error.

```
// load sample.tga into image
SDL_Surface *image;
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.tga", "rb");
image=IMG_LoadTGA_RW(rwop);
if(!image) {
    printf("IMG_LoadTGA_RW: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10

3.1.14 IMG_LoadLBM_RW

`SDL_Surface *`**IMG_LoadLBM_RW**`(SDL_RWops *`*src*`)`

src The source SDL_RWops as a pointer. The LBM image is loaded from this.

Load *src* as a LBM image for use as a surface, if LBM support is compiled into the SDL_image library.

Returns: a pointer to the image as a new `SDL_Surface`. **NULL** is returned on errors, like if LBM is not supported, or a read error.

```
// load sample.lbm into image
SDL_Surface *image;
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.lbm", "rb");
image=IMG_LoadLBM_RW(rwop);
if(!image) {
    printf("IMG_LoadLBM_RW: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10, Section 3.2.10 [IMG_isLBM], page 33

3.1.15 IMG_ReadXPMFromArray

SDL_Surface *IMG_ReadXPMFromArray(char **xpm)

xpm The source xpm data. The XPM image is loaded from this. XPM files are C header files that define a char **variable, that variable name is what you use here.

Load *xpm* as a XPM image for use as a surface, if XPM support is compiled into the SDL_image library.

Returns: a pointer to the image as a new SDL_Surface. **NULL** is returned on errors, like if XPM is not supported, or a read error.

```
// load sample.xpm into image
#include "sample.xpm"
SDL_Surface *image;
image=IMG_ReadXPMFromArray(sample_xpm);
if(!image) {
    printf("IMG_ReadXPMFromArray: %s\n", IMG_GetError());
    // handle error
}
```

See Also:
Section 3.1.6 [IMG_LoadXPM_RW], page 13

3.2 Info

These functions are tests for specific file formats. They also show if the format is supported in the linked SDL_image library.

3.2.1 IMG_isBMP

`int` **IMG_isBMP**`(SDL_RWops *`*`src`*`)`

src

If the BMP format is supported, then the image data is tested to see if it is readable as a BMP, otherwise it returns false (Zero).

Returns: 1 if the image is a BMP and the BMP format support is compiled into SDL_image. 0 is returned otherwise.

```
// Test sample.bmp to see if it is a BMP
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.bmp", "rb");
if(IMG_isBMP(rwop))
printf("sample.bmp is a BMP file.\n");
else
printf("sample.bmp is not a BMP file, or BMP support is not available.\n");
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10

3.2.2 IMG_isPNM

`int` **IMG_isPNM**`(SDL_RWops *`*`src`*`)`

src

If the PNM format is supported, then the image data is tested to see if it is readable as a PNM, otherwise it returns false (Zero).

Returns: 1 if the image is a PNM and the PNM format support is compiled into SDL_image. 0 is returned otherwise.

```
// Test sample.pnm to see if it is a PNM
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.pnm", "rb");
if(IMG_isPNM(rwop))
printf("sample.pnm is a PNM file.\n");
else
printf("sample.pnm is not a PNM file, or PNM support is not available.\n");
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10

3.2.3 IMG_isXPM

int **IMG_isXPM**(SDL_RWops *src)

src

If the XPM format is supported, then the image data is tested to see if it is readable as a XPM, otherwise it returns false (Zero).

Returns: 1 if the image is a XPM and the XPM format support is compiled into SDL_image. 0 is returned otherwise.

```
// Test sample.xpm to see if it is a XPM
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.xpm", "rb");
if(IMG_isXPM(rwop))
printf("sample.xpm is a XPM file.\n");
else
printf("sample.xpm is not a XPM file, or XPM support is not available.\n");
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10

3.2.4 IMG_isXCF

`int` **IMG_isXCF**`(SDL_RWops *`*`src`*`)`

src

If the XCF format is supported, then the image data is tested to see if it is readable as a XCF, otherwise it returns false (Zero).

Returns: 1 if the image is a XCF and the XCF format support is compiled into SDL_image. 0 is returned otherwise.

```
// Test sample.xcf to see if it is a XCF
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.xcf", "rb");
if(IMG_isXCF(rwop))
printf("sample.xcf is a XCF file.\n");
else
printf("sample.xcf is not a XCF file, or XCF support is not available.\n");
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10

3.2.5 IMG_isPCX

`int` **IMG_isPCX**`(SDL_RWops *`*`src`*`)`

src

If the PCX format is supported, then the image data is tested to see if it is readable as a PCX, otherwise it returns false (Zero).

Returns: 1 if the image is a PCX and the PCX format support is compiled into SDL_image. 0 is returned otherwise.

```
// Test sample.pcx to see if it is a PCX
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.pcx", "rb");
if(IMG_isPCX(rwop))
printf("sample.pcx is a PCX file.\n");
else
printf("sample.pcx is not a PCX file, or PCX support is not available.\n");
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10

3.2.6 IMG_isGIF

int **IMG_isGIF**(SDL_RWops **src*)

src

If the GIF format is supported, then the image data is tested to see if it is readable as a GIF, otherwise it returns false (Zero).

Returns: 1 if the image is a GIF and the GIF format support is compiled into SDL_image. 0 is returned otherwise.

```
// Test sample.gif to see if it is a GIF
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.gif", "rb");
if(IMG_isGIF(rwop))
printf("sample.gif is a GIF file.\n");
else
printf("sample.gif is not a GIF file, or GIF support is not available.\n");
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10

3.2.7 IMG_isJPG

`int IMG_isJPG(SDL_RWops *src)`

src

If the JPG format is supported, then the image data is tested to see if it is readable as a JPG, otherwise it returns false (Zero).

Returns: 1 if the image is a JPG and the JPG format support is compiled into SDL_image. 0 is returned otherwise.

```
// Test sample.jpg to see if it is a JPG
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.jpg", "rb");
if(IMG_isJPG(rwop))
printf("sample.jpg is a JPG file.\n");
else
printf("sample.jpg is not a JPG file, or JPG support is not available.\n");
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10

3.2.8 IMG_isTIF

int **IMG_isTIF**(SDL_RWops *src)

src

If the TIF format is supported, then the image data is tested to see if it is readable as a TIF, otherwise it returns false (Zero).

Returns: 1 if the image is a TIF and the TIF format support is compiled into SDL_image. 0 is returned otherwise.

```
// Test sample.tif to see if it is a TIF
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.tif", "rb");
if(IMG_isTIF(rwop))
printf("sample.tif is a TIF file.\n");
else
printf("sample.tif is not a TIF file, or TIF support is not available.\n");
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10

3.2.9 IMG_isPNG

`int` **IMG_isPNG**`(SDL_RWops *`*`src`*`)`

src

If the PNG format is supported, then the image data is tested to see if it is readable as a PNG, otherwise it returns false (Zero).

Returns: 1 if the image is a PNG and the PNG format support is compiled into SDL_image. 0 is returned otherwise.

```
// Test sample.png to see if it is a PNG
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.png", "rb");
if(IMG_isPNG(rwop))
printf("sample.png is a PNG file.\n");
else
printf("sample.png is not a PNG file, or PNG support is not available.\n");
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10

3.2.10 IMG_isLBM

int **IMG_isLBM**(SDL_RWops **src*)

src

If the LBM format is supported, then the image data is tested to see if it is readable as a LBM, otherwise it returns false (Zero).

Returns: 1 if the image is a LBM and the LBM format support is compiled into SDL_image. 0 is returned otherwise.

```
// Test sample.lbm to see if it is a LBM
SDL_RWops *rwop;
rwop=SDL_RWFromFile("sample.lbm", "rb");
if(IMG_isLBM(rwop))
printf("sample.lbm is a LBM file.\n");
else
printf("sample.lbm is not a LBM file, or LBM support is not available.\n");
```

See Also:
Section 3.1.3 [IMG_LoadTyped_RW], page 10

3.3 Errors

These functions are used for error status strings that should help the user and developer understand why a function failed.

3.3.1 IMG_SetError

`void` **IMG_SetError**`(const char *`*fmt*`, ...)`

This is the same as SDL_SetError, which sets the error string which may be fetched with IMG_GetError (or SDL_GetError). This functions acts like printf, except that it is limited to **SDL_ERRBUFIZE**(1024) chars in length. It only accepts the following format types: `%s, %d, %f, %p`. No variations are supported, like `%.2f` would not work. For any more specifics read the SDL docs.

```
int myimagefunc(int i) {
    IMG_SetError("myimagefunc is not implemented! %d was passed in.",i);
    return(-1);
}
```

See Also:
Section 3.3.2 [IMG_GetError], page 36

3.3.2 IMG_GetError

char *IMG_GetError()

This is the same as SDL_GetError, which returns the last error set as a string which you may use to tell the user what happened when an error status has been returned from an SDL_image function call.

Returns: a char pointer (string) containing a humam readble version or the reason for the last error that occured.

```
printf("Oh My Goodness, an error : %s", IMG_GetError());
```

See Also:
Section 3.3.1 [IMG_SetError], page 35

Index