

Table of Contents

1 Overview.....	1
2 History.....	2
3 License.....	3
4 Structures.....	4
4.1 GFL_BITMAP.....	4
4.2 GFL_COLORMAP.....	5
4.3 GFL_FORMAT_INFORMATION.....	5
4.4 GFL_FILE_INFORMATION.....	6
4.5 GFL_LOAD_PARAMS.....	8
4.6 GFL_SAVE_PARAMS.....	10
4.7 GFL_RECT.....	12
4.8 GFL_COLOR.....	12
4.9 GFL_POINT.....	13
4.10 GFL_FILTER.....	13
4.11 GFL_IPTC_DATA.....	13
4.12 GFL_EXIF_DATA.....	14
4.13 GFL_LOAD_CALLBACKS.....	15
4.14 GFL_EXIF_DATAEX.....	15

1 Overview

GFL SDK is a powerfull library for developers who would like to support graphics image formats easily.

- Support more than 100 graphics formats.
- 8 or 16 bits per component
- Color depth conversion
- LUT transforms
- Filters
- Multi-page creation
- JPEG lossless rotation
- EXIF/IPTC reading
- IPTC writing
- Multithreading
- C++, VB, Delphi wrapper
- Windows, Linux, MacOS X, Irix, Solaris, FreeBSD, BeOS
- ...

2 History

GFL v2.54 (LIBFORMAT v4.57) 15/03/2006:

Fixed : Many bugs & improved

Added : in GFL_LOAD_PARAMS, PNGComposeWithAlpha & WMFHighResolution

Added : gflReduceNoise

Added : gflSetTransparentColor

Fixed : Many bugs & improved speed

3 License

Installing and using this library (GFL SDK/GfIAX) signifies acceptance of these terms and conditions of the license.

- "GFL SDK/GFLAx" is provided as Freeware for private non-commercial or educational use (including non-profit organization). You must contact me for commercial use and distribution. webmaster@xnview.com
- You may not use GFL SDK or GfIAX to create components or controls to be used by other developers without written approval.
- The product developed by the Licensee should not be similar to or should not compete with XnView/NConvert (should not be a graphic viewer or converter).
- "GFL SDK/GFLAx" IS NOT DESIGNED, MANUFACTURED OR INTENDED FOR USE OR RESALE AS ONLINE CONTROL EQUIPMENT IN HAZARDOUS ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS IN THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATIONS SYSTEMS, AIR TRAFFIC CONTROL, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH THE FAILURE OF "GFL SDK/GFLAx" COULD LEAD DIRECTLY TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE.
- "GFL SDK/GFLAx" is provided "as-is" and without warranty of any kind, express, implied or otherwise, including without limitation, any warranty of merchantability or fitness for a particular purpose.
- In no event shall the author of this software be held liable for data loss, damages, loss of profits or any other kind of loss while using or misusing this software.
- You may not use, copy, emulate, clone, rent, lease, sell, modify, decompile, disassemble, otherwise reverse engineer, or transfer the licensed program, or any subset of the licensed program, except as provided for in this agreement. Any such unauthorized use shall result in immediate and automatic termination of this license and may result in criminal and/or civil prosecution.
- Important For JPEG-2000 & JBIG use, see corresponding licenses in Plugins folder of the GFL SDK package

Any suggestions, feedback and comments are welcome.

4 Structures

4.1 GFL_BITMAP

The GFL_BITMAP structure contains all informations about picture in memory.

```
struct GFL_BITMAP (
    GFL_BITMAP_TYPE Type
    GFL_ORIGIN Origin
    GFL_INT32 Width
    GFL_INT32 Height
    GFL_UINT32 BytesPerLine
    GFL_INT16 LinePadding
    GFL_UINT16 BitsPerComponent
    GFL_UINT16 ComponentsPerPixel
    GFL_UINT16 BytesPerPixel
    GFL_UINT16 Xdpi
    GFL_UINT16 Ydpi
    GFL_INT16 TransparentIndex
    GFL_INT16 Reserved
    GFL_INT32 ColorUsed
    GFL_COLOMAP* ColorMap
    GFL_UINT8* Data
    char* Comment
    void* MetaData
    GFL_INT32 XOffset
    GFL_INT32 YOffset
    char* Name
}
```

Members

Type

Type of the picture

GFL_BINARY	0x0001	Binary
GFL_GREY	0x0002	Grey scale
GFL_COLORS	0x0004	Colors with colormap
GFL_RGB	0x0010	TrueColors - Red/Green/Blue
GFL_RGBA	0x0020	TrueColors - Red/Green/Blue/Alpha
GFL_BGR	0x0040	TrueColors - Blue/Green/Red
GFL_ABGR	0x0080	TrueColors - Alpha/Blue/Green/Red
GFL_BGRA	0x0100	TrueColors - Blue/Green/Red/Alpha
GFL_ARGB	0x0200	TrueColors - Alpha/Red/Green/Blue
GFL_CMYK	0x0400	TrueColors - Cyan/Magenta/Yellow/Black

Origin

Origin of the picture.

GFL_TOP_LEFT	0	Top left (default)
GFL_BOTTOM_LEFT	2	Bottom left
GFL_TOP_RIGHT	1	Top right
GFL_BOTTOM_RIGHT	3	Bottom right

Width

Width	Width in pixels of the picture.
Height	Height in pixels of the picture.
BytesPerLine	Bytes per line of pixels.
LinePadding	Internal use, do not modify.
BitsPerComponent	Bits per component, can be 1, 8, 16
ComponentsPerPixel	Component per pixel, can be 1, 3 or 4
BytesPerPixel	Bytes per pixel (For example: 1, 3 or 4).
Xdpi	Pixels per inch in X axis.
Ydpi	Pixels per inch in Y axis.
TransparentIndex	Index of transparency (only for GFL_COLORS & GFL_GREY type).
ColorUsed	Number of color used in the picture (only for GFL_COLORS & GFL_GREY type).
ColorMap	Address of a GFL_COLORMAP structure for the colormap (only for GFL_COLORS type).
Data	Pointer of the picture data.
Comment	Address of a string used by the comment. You must use <code>gflSetComment</code> to change the comment.
MetaData	Pointer of Metadata. You must use <code>gflBitmapGetIPTC</code> & <code>gflBitmapGetEXIF</code> to obtain readable data.

4.2 GFL_COLORMAP

The GFL_COLORMAP structure is used for colormap.

```
struct GFL_COLORMAP (
    GFL_UINT8 Red[256]
    GFL_UINT8 Green[256]
    GFL_UINT8 Blue[256]
)
```

Members

Red	
	Array of red components.
Green	
	Array of green components.
Blue	
	Array of blue components.

4.3 GFL_FORMAT_INFORMATION

The GFL_FORMAT_INFORMATION structure contains informations about a format available in GFL.

```
struct GFL_FORMAT_INFORMATION (
    GFL_INT32 Index
    GFL_ORIGIN Name[8]
    char Description[64]
    GFL_UINT32 Status
)
```

```

    GFL_UINT32 NumberOfExtension
    char Extension[16][8]
}

```

Members

Index

Index of the format.

Name

Null-terminated string that contains the name of the format. For example, "jpeg" is for JPEG format.

Description

Null-terminated string that contains the label of the format.

Status

Format status.

GFL_READ Reading support

GFL_WRITE Writing support

NumberOfExtension

Number of extension known by this format.

Extension

Array of Null-terminated string that contains the extension.

4.4 GFL_FILE_INFORMATION

The GFL_FILE_INFORMATION structure contains informations about a picture's file.

```

struct GFL_FILE_INFORMATION (
    GFL_BITMAP_TYPE Type
    GFL_ORIGIN Origin
    GFL_INT32 Width
    GFL_INT32 Height
    GFL_INT32 FormatIndex
    char FormatName[8]
    char Description[64]
    GFL_UINT16 Xdpi
    GFL_UINT16 Ydpi
    GFL_UINT16 BitsPerComponent
    GFL_UINT16 ComponentsPerPixel
    GFL_INT32 NumberOfImages
    GFL_UINT32 FileSize
    GFL_COLORMODEL ColorModel
    GFL_COMPRESSION Compression
    char CompressionDescription[64]
    GFL_INT32 XOffset
    GFL_INT32 YOffset
    void* ExtraInfos
}

```

Members

Type

Not used

Origin

Origin of the picture.

GFL_TOP_LEFT 0 Top left (default)

GFL_BOTTOM_LEFT 2 Bottom left

GFL_TOP_RIGHT 1 Top right

GFL_BOTTOM_RIGHT 3 Bottom right

Width

Width in pixels of the picture.

Height

Height in pixels of the picture.

FormatIndex

Index of picture's format.

FormatName

Name of picture's format.

Description

File label.

Xdpi

Pixels per inch in the X axis.

Ydpi

Pixels per inch in the Y axis.

BitsPerComponent

Bits per component, can be 1, 8, 16

ComponentsPerPixel

Component per pixel, can be 1, 3 or 4

NumberOfImages

Number of picture in the file.

FileSize

Size of the file.

ColorModel

Color model.

GFL_CM_RGB	0	Red-Green-Blue
GFL_CM_GREY	1	Greyscale
GFL_CM_CMY	2	Cyan-Magenta-Yellow
GFL_CM_CMYK	3	Cyan-Magenta-Yellow-Black
GFL_CM_YCBCR	4	YCbCr
GFL_CM_YUV16	5	YUV 16bits
GFL_CM_LAB	6	Lab
GFL_CM_LOGLUV	7	Log Luv
GFL_CM_LOGL	8	Log L

Compression

GFL_NO_COMPRESSION	0	No compression
GFL_RLE	1	Packbits
GFL_LZW	2	LZW
GFL_JPEG	3	JPEG
GFL_ZIP	4	ZIP
GFL_SGI_RLE	5	SGI Packbits
GFL_CCITT_RLE	6	CCITT RLE
GFL_CCITT_FAX3	7	Fax Group 3
GFL_CCITT_FAX3_2D	8	Fax Group 3-2D
GFL_CCITT_FAX4	9	Fax Group 4
GFL_WAVELET	10	Wavelet
GFL_UNKNOWN_COMPRESSION	255	Other compression

CompressionDescription

Pointer to a buffer that contains the full compression description.

Remarks

gflFreeFileInformation must be used for freeing the allocated memory.

4.5 GFL_LOAD_PARAMS

The GFL_LOAD_PARAMS structure contains options for picture loading.

```
struct GFL_LOAD_PARAMS (
    GFL_UINT32 Flags
    GFL_INT32 FormatIndex
    GFL_INT32 ImageWanted
    GFL_ORIGIN Origin
    GFL_BITMAP_TYPE ColorModel
    GFL_UINT32 LinePadding
    GFL_UINT8 DefaultAlpha
    GFL_UINT8 PsdNoAlphaForNonLayer
    GFL_UINT8 PngComposeWithAlpha
    GFL_UINT8 WMFHighResolution
    GFL_INT32 Width
    GFL_INT32 Height
    GFL_UINT32 Offset
    GFL_CORDER ChannelOrder
    GFL_CTYPE ChannelType
    GFL_UINT16 PcdBase
    GFL_UINT16 EpsDpi
    GFL_INT32 EpsWidth
    GFL_INT32 EpsHeight
    GFL_LUT_TYPE LutType
    GFL_UINT16 Reserved
    GFL_UINT16* LutData
    const char* LutFilename
    GFL_UINT8 CameraRawUseAutomaticBalance
    GFL_UINT8 CameraRawUseCameraBalance
    GFL_UINT8 CameraRawHighlight
    GFL_UINT8 CameraRawAutoBright
    float CameraRawGamma
    float CameraRawBrightness
    float CameraRawRedScaling
    float CameraRawBlueScaling
    GFL_LOAD_CALLBACKS Callbacks
    void* UserParams
)
```

Members

Flags

Options

GFL_LOAD_SKIP_ALPHA	If the picture has an alpha channel, it is ignored
GFL_LOAD_IGNORE_READ_ERROR	Ignore all read errors
GFL_LOAD_BY_EXTENSION_ONLY	Use only extension to recognize the filetype
GFL_LOAD_READ_ALL_COMMENT	Read all comment in the file
GFL_LOAD_FORCE_COLOR_MODEL	ColorModel is used for the picture type
GFL_LOAD_PREVIEW_NO_CANVAS_RESIZE	Keep the ratio for the preview
GFL_LOAD_BINARY_AS_GREY	Load a binary file in 8bits
GFL_LOAD_ORIGINAL_COLORMODEL	If the color model of the file is CMYK, so the picture loaded will be in CMYK
GFL_LOAD_ONLY_FIRST_FRAME	If the color model of the file is CMYK, so the picture loaded will be in CMYK

GFL_LOAD_ORIGINAL_DEPTH	If the file has more than 8 bits per component, keep it
GFL_LOAD_METADATA	Read all metadata (IPTC & EXIF)
GFL_LOAD_COMMENT	Read comment
GFL_LOAD_HIGH_QUALITY_THUMBNAIL	Use high quality for gflLoadThumbnail

FormatIndex

Index of the format used to load.
Default value : -1 (for an automatic recognition).

ImageWanted

For a multi-page file, identifies the image number.
Default value : 0

Origin

Origin wanted.

GFL_TOP_LEFT	Top left
GFL_BOTTOM_LEFT	Bottom left
GFL_TOP_RIGHT	Top right
GFL_BOTTOM_RIGHT	Bottom right

Default value : GFL_TOP_LEFT

ColorModel

Color Model wanted.

GFL_RGB	True colors - Red/Green/Blue (24 bits)
GFL_BGR	True colors - Blue/Green/Red (24 bits)
GFL_RGBA	True colors - Red/Green/Blue/Alpha (32 bits)
GFL_ABGR	True colors - Alpha/Blue/Green/Red (32 bits)
GFL_BGRA	True colors - Blue/Green/Red/Alpha (32 bits)
GFL_ARGB	True colors - Red/Green/Blue/Alpha (32 bits)

Default value : GFL_RGB

LinePadding

Pad for a pixels line (For example, a value of 4 allow a line padding on 32bits).
Default value : 1

DefaultAlpha

Alpha value to use when the picture is loaded in 32bits, but the original file doesn't have an alpha.
Default value: Black

Width

For RAW or YUV format, width of picture.

Height

For RAW or YUV format, height of picture.

Offset

For RAW or YUV format, offset of the picture in the file.

ChannelOrder

For RAW format, channel order of the components. GFL_CORDER_INTERLEAVED :Interleaved

GFL_CORDER_SEQUENTIAL	Sequential
GFL_CORDER_SEPARATE	Separate

ChannelType

For RAW format, channel type of the components.

GFL_CTYPE_GREYSCALE	Greyscale
---------------------	-----------

GFL_CTYPE_RGB	Red-Green-Blue
GFL_CTYPE_BGR	Blue-Green-Red
GFL_CTYPE_RGBA	Red-Green-Blue-Alpha
GFL_CTYPE_ABGR	Alpha-Blue-Green-Red
GFL_CTYPE_CMY	Cyan-Magenta-Yellow
GFL_CTYPE_CMYK	Cyan-Magenta-Yellow-Black

PcdBase

For PCD format, it's the base used.

0 192x144
1 384x288
2 768x576

EpsDpi

For PS/EPS format, dpi to be used for loading.

EpsWidth

For PS/EPS format, width to be used for loading.

EpsHeight

For PS/EPS format, height to be used for loading.

Callbacks

A **GFL_LOAD_CALLBACKS** structure

4.6 GFL_SAVE_PARAMS

The GFL_SAVE_PARAMS structure contains options for the save of picture.

```
struct GFL_SAVE_PARAMS (
    GFL_UINT32 Flags
    GFL_INT32 FormatIndex
    GFL_COMPRESSION Compression
    GFL_INT16 Quality
    GFL_INT16 CompressionLevel
    GFL_BOOL Interlaced
    GFL_BOOL Progressive
    GFL_BOOL OptimizeHuffmanTable
    GFL_BOOL InAscii
    GFL_LUT_TYPE LutType
    GFL_UINT8 DpxByteOrder
    GFL_UINT8 CompressRatio
    GFL_UINT32 MaxFileSize
    GFL_UINT16* LutData
    const char* LutFilename
    GFL_UINT32 Offset
    GFL_CORDER ChannelOrder
    GFL_CTYPE ChannelType
    GFL_WRITE_CALLBACK Write
    GFL_TELL_CALLBACK Tell
    GFL_SEEK_CALLBACK Seek
    GFL_VIRTUAL_SAVE_CALLBACK GetLine
    void* GetLineParams
    void* UserParams
)
```

Members

Flags

Options

GFL_SAVE_REPLACE_EXTENSION Replace extension by the default format extension

GFL_SAVE_ANYWAY Convert picture if colormode can be saved in this format (For example, RGB picture must be converted in 256 colors to save it in GIF)

FormatIndex

Index of format to be used.

Compression

GFL_NO_COMPRESSION No compression

GFL_RLE Packbits

GFL_LZW LZW (tiff only)

GFL_CCITT_FAX3 Fax Group 3 (tiff only)

GFL_CCITT_FAX3_2D Fax Group 3-2D (tiff only)

GFL_CCITT_FAX4 Fax Group 4 (tiff only)

Quality

Quality of the compression (JPEG)

0: the worst, 100: the best

CompressionLevel

Level of compression (PNG).

1: minimum, 7: maximum

Interlaced

Interlaced mode (GIF).

Progressive

Progressive mode (JPEG).

OptimizeHuffmanTable

Optimize the Huffman table (JPEG).

InAscii

Use the ascii mode (PNM)

Offset

For RAW or YUV format, offset of the data start.

ChannelOrder

For RAW format, channel order of components.

GFL_CORDER_INTERLEAVED Interleaved

GFL_CORDER_SEQUENTIAL Sequential

GFL_CORDER_SEPARATE Separate

ChannelType

For RAW format, channel type of components.

GFL_CTYPE_GREYSCALE Greyscale

GFL_CTYPE_RGB Red-Green-Blue

GFL_CTYPE_BGR Bleu-Green-Red

GFL_CTYPE_RGBA Red-Green-Bleu-Alpha

GFL_CTYPE_ABGR Alpha-Bleu-Green-Red

GFL_CTYPE_CMY Cyan-Magenta-Yellow

GFL_CTYPE_CMYK Cyan-Magenta-Yellow-Black

Write

	Pointer to a write user function.
Tell	Pointer to a tell user function.
Seek	Pointer to a seek user function.

4.7 GFL_RECT

The GFL_RECT structure define a rectangle.

```
struct GFL_RECT(
    GFL_INT32 x
    GFL_INT32 y
    GFL_INT32 w
    GFL_INT32 h
)
```

Members

x	
	X position.
y	
	Y position.
w	
	Width.
h	
	Height.

4.8 GFL_COLOR

The GFL_COLOR structure allow to define a color.

```
struct GFL_COLOR (
    GFL_UINT16 Red
    GFL_UINT16 Green
    GFL_UINT16 Blue
    GFL_UINT16 Alpha
)
```

Members

Red	
	Define the red component.
Green	
	Define the green component.
Blue	
	Define the blue component.
Alpha	
	Define the alpha component.

4.9 GFL_POINT

The GFL_POINT structure allows to define a point.

```
struct GFL_POINT {  
    GFL_INT32 x  
    GFL_INT32 y  
}
```

Members

x	
	X position.
y	
	Y position.

4.10 GFL_FILTER

The GFL_FILTER structure allows to define a matrix for convolution (maximum 7x7).

```
struct GFL_FILTER {  
    GFL_INT16 Size  
    GFL_INT16 Matrix[7*7]  
    GFL_INT16 Divisor  
    GFL_INT16 Bias  
}
```

Members

Size	
	Define the width of the matrix (maximum 7).
Matrix	
	Define each values fo the matrix.
Divisor	
	Define the divisor to apply.
Bias	
	Define the bias to apply.

Example

A "blur" matrix is defined like this:

```
Size = 3  
Matrix = (1 2 1 2 4 2 1 2 1)  
Divisor = 16  
Bias = 0
```

4.11 GFL_IPTC_DATA

```
struct GFL_IPTC_ENTRY {  
    GFL_UINT32 Id  
    const char* Name  
    const char* Value  
}
```

Members

Id	
	ID of the field
Name	
	Pointer to a null-terminated string of the field's name
Value	
	Pointer to a null-terminated string of the field's value

```
struct GFL_IPTC_DATA {
    GFL_UINT32      NumberOfItems
    GFL_IPTC_ENTRY* ItemsList
}
```

Members

NumberOfItems	
	Number of iptc fields
ItemsList	
	List of iptc fields

4.12 GFL_EXIF_DATA

IFD

GFL_EXIF_IFD_0	0x0001
GFL_EXIF_MAIN_IFD	0x0002
GFL_EXIF_INTEROPERABILITY_IFD	0x0004
GFL_EXIF_IFD_THUMBNAIL	0x0008
GFL_EXIF_GPS_IFD	0x0010
GFL_EXIF_MAKERNOTE_IFD	0x0020

Some tags

GFL_EXIF_MAKER	0x010F
GFL_EXIF_MODEL	0x0110
GFL_EXIF_ORIENTATION	0x0112
GFL_EXIF_EXPOSURETIME	0x829A
GFL_EXIF_FNUMBER	0x829D
GFL_EXIF_DATETIME_ORIGINAL	0x9003
GFL_EXIF_SHUTTERSPEED	0x9201
GFL_EXIF_APERTURE	0x9202
GFL_EXIF_MAXAPERTURE	0x9205
GFL_EXIF_FOCALLENGTH	0x920A

```
struct GFL_EXIF_ENTRY {
    GFL_UINT32  Flag /* EXIF_...IFD */
    GFL_UINT32  Tag
    const char* Name
    const char* Value
}
```

```
struct GFL_EXIF_DATA {
```

```

    GFL_UINT32      NumberOfItems
    GFL_EXIF_ENTRY* ItemsList
}

```

4.13 GFL_LOAD_CALLBACKS

```

struct GFL_LOAD_CALLBACKS {
    GFL_READ_CALLBACK Read
    GFL_TELL_CALLBACK Tell
    GFL_SEEK_CALLBACK Seek
    GFL_ALLOCATEBITMAP_CALLBACK AllocateBitmap
    void* AllocateBitmapParams
    GFL_PROGRESS_CALLBACK Progress
    void* ProgressParams
    GFL_WANTCANCEL_CALLBACK WantCancel
    void* WantCancelParams
    GFL_VIRTUAL_LOAD_CALLBACK SetLine
    void* SetLineParams
}

```

4.14 GFL_EXIF_DATAEX

Tag format

GFL_EXIF_BYTE	1
GFL_EXIF_STRING	2
GFL_EXIF_USHORT	3
GFL_EXIF_ULONG	4
GFL_EXIF_URATIONAL	5
GFL_EXIF_SBYTE	6
GFL_EXIF_UNDEFINED	7
GFL_EXIF_SSHORT	8
GFL_EXIF_SLONG	9
GFL_EXIF_SRATIONAL	10
GFL_EXIF_SINGLEF	11
GFL_EXIF_DOUBLE	12

```

struct GFL_EXIF_ENTRYEX {
    GFL_UINT16 Tag
    GFL_UINT16 Format
    GFL_INT32 Ifd
    GFL_INT32 NumberOfComponents
    GFL_UINT32 Value
    GFL_INT32 DataLength
    char* Data
    GFL_EXIF_ENTRYEX* Next
}

```

```

struct GFL_EXIF_DATAEX {
    GFL_ENTRYEX* Root
    GFL_INT32 UseMsbf
}

```