# INTRO

[NAME](#gjdgxs)

[SYNOPSIS](#30j0zll)

[DESCRIPTION](#1fob9te)

[DATA TYPES](#3znysh7)

[LIST OF ROUTINES](#2et92p0)

[TAG USAGE](#tyjcwt)

[PSEUDO TAGS](#3dy6vkm)

[DIAGNOSTICS](#1t3h5sf)

[SEE ALSO](#4d34og8)

[BUGS](#2s8eyo1)

## NAME

|  |  |
| --- | --- |
|  | libtiff − introduction to *libtiff*, a library for reading and writing TIFF files |

## SYNOPSIS

|  |  |
| --- | --- |
|  | **#include <tiffio.h>**  cc file.c **-ltiff** |

## DESCRIPTION

|  |  |
| --- | --- |
|  | *libtiff* is a library for reading and writing data files encoded with the *Tag Image File* format, Revision 6.0 (or revision 5.0 or revision 4.0). This file format is suitable for archiving multi-color and monochromatic image data.  The library supports several compression algorithms, as indicated by the *Compression* field, including: no compression (1), CCITT 1D Huffman compression (2), CCITT Group 3 Facsimile compression (3), CCITT Group 4 Facsimile compression (4), Lempel-Ziv & Welch compression (5), baseline JPEG compression (7), word-aligned 1D Huffman compression (32771), and PackBits compression (32773). In addition, several nonstandard compression algorithms are supported: the 4-bit compression algorithm used by the *ThunderScan* program (32809) (decompression only), NeXT’s 2-bit compression algorithm (32766) (decompression only), an experimental LZ-style algorithm known as Deflate (32946), and an experimental CIE LogLuv compression scheme designed for images with high dynamic range (32845 for LogL and 32845 for LogLuv). Directory information may be in either little- or big-endian byte order−byte swapping is automatically done by the library. Data bit ordering may be either Most Significant Bit ( MSB ) to Least Significant Bit ( LSB ) or LSB to MSB. Finally, the library does not support files in which the *BitsPerSample*, *Compression*, *MinSampleValue*, or *MaxSampleValue* fields are defined differently on a per-sample basis (in Rev. 6.0 the *Compression* tag is not defined on a per-sample basis, so this is immaterial). |

## DATA TYPES

|  |  |
| --- | --- |
|  | The library makes extensive use of C typedefs to promote portability. Two sets of typedefs are used, one for communication with clients of the library and one for internal data structures and parsing of the TIFF format. The following typedefs are exposed to users either through function definitions or through parameters passed through the varargs interfaces. |

|  |  |  |
| --- | --- | --- |
|  | typedef unsigned short uint16; | 16-bit unsigned integer |
|  | typedef unsigned <*thing*> uint32; | 32-bit unsigned integer |
|  | typedef unsigned int ttag\_t; | directory tag |
|  | typedef uint16 tdir\_t; | directory index |
|  | typedef uint16 tsample\_t; | sample number |
|  | typedef uint32 tstrip\_t; | strip number |
|  | typedef uint32 ttile\_t; | tile number |
|  | typedef int32 tsize\_t; | i/o size in bytes |
|  | typedef void\* tdata\_t; | image data ref |
|  | typedef void\* thandle\_t; | client data handle |
|  | typedef int32 toff\_t; | file offset |

|  |  |
| --- | --- |
|  | Note that *tstrip\_t*, *ttile\_t*, and *tsize\_t* are constrained to be no more than 32-bit quantities by 32-bit fields they are stored in in the TIFF image. Likewise *tsample\_t* is limited by the 16-bit field used to store the *SamplesPerPixel* tag. *tdir\_t* constrains the maximum number of IFDs that may appear in an image and may be an arbitrary size (w/o penalty). *ttag\_t* must be either int, unsigned int, pointer, or double because the library uses a varargs interface and ANSI C restricts the type of the parameter before an ellipsis to be a promoted type. *toff\_t* is defined as int32 because TIFF file offsets are (unsigned) 32-bit quantities. A signed value is used because some interfaces return −1 on error. Finally, note that user-specified data references are passed as opaque handles and only cast at the lowest layers where their type is presumed. |

## LIST OF ROUTINES

|  |  |
| --- | --- |
|  | The following routines are part of the library. Consult specific manual pages for details on their operation; on most systems doing ‘‘man function-name’’ will work. |

*Name Description*

|  |  |  |
| --- | --- | --- |
|  | TIFFCheckpointDirectory | writes the current state of the directory |
|  | TIFFCheckTile | very x,y,z,sample is within image |
|  | TIFFCIELabToRGBInit | initialize CIE L\*a\*b\* 1976 to RGB conversion state |
|  | TIFFCIELabToXYZ | perform CIE L\*a\*b\* 1976 to CIE XYZ conversion |
|  | TIFFClientOpen | open a file for reading or writing |
|  | TIFFClose | close an open file |
|  | TIFFComputeStrip | return strip containing y,sample |
|  | TIFFComputeTile | return tile containing x,y,z,sample |
|  | TIFFCurrentDirectory | return index of current directory |
|  | TIFFCurrentRow | return index of current scanline |
|  | TIFFCurrentStrip | return index of current strip |
|  | TIFFCurrentTile | return index of current tile |
|  | TIFFDataWidth | return the size of TIFF data types |
|  | TIFFError | library error handler |
|  | TIFFFdOpen | open a file for reading or writing |
|  | TIFFFileName | return name of open file |
|  | TIFFFileno | return open file descriptor |
|  | TIFFFindCODEC | find standard codec for the specific scheme |
|  | TIFFFlush | flush all pending writes |
|  | TIFFFlushData | flush pending data writes |
|  | TIFFGetBitRevTable | return bit reversal table |
|  | TIFFGetField | return tag value in current directory |
|  | TIFFGetFieldDefaulted | return tag value in current directory |
|  | TIFFGetMode | return open file mode |
|  | TIFFGetVersion | return library version string |
|  | TIFFIsCODECConfigured | check, whether we have working codec |
|  | TIFFIsMSB2LSB | return true if image data is being returned |
|  |  | with bit 0 as the most significant bit |
|  | TIFFIsTiled | return true if image data is tiled |
|  | TIFFIsByteSwapped | return true if image data is byte-swapped |
|  | TIFFNumberOfStrips | return number of strips in an image |
|  | TIFFNumberOfTiles | return number of tiles in an image |
|  | TIFFOpen | open a file for reading or writing |
|  | TIFFPrintDirectory | print description of the current directory |
|  | TIFFReadBufferSetup | specify i/o buffer for reading |
|  | TIFFReadDirectory | read the next directory |
|  | TIFFReadEncodedStrip | read and decode a strip of data |
|  | TIFFReadEncodedTile | read and decode a tile of data |
|  | TIFFReadRawStrip | read a raw strip of data |
|  | TIFFReadRawTile | read a raw tile of data |
|  | TIFFReadRGBAImage | read an image into a fixed format raster |
|  | TIFFReadScanline | read and decode a row of data |
|  | TIFFReadTile | read and decode a tile of data |
|  | TIFFRegisterCODEC | override standard codec for the specific scheme |
|  | TIFFReverseBits | reverse bits in an array of bytes |
|  | TIFFRGBAImageBegin | setup decoder state for TIFFRGBAImageGet |
|  | TIFFRGBAImageEnd | release TIFFRGBAImage decoder state |
|  | TIFFRGBAImageGet | read and decode an image |
|  | TIFFRGBAImageOK | is image readable by TIFFRGBAImageGet |
|  | TIFFScanlineSize | return size of a scanline |
|  | TIFFSetDirectory | set the current directory |
|  | TIFFSetSubDirectory | set the current directory |
|  | TIFFSetErrorHandler | set error handler function |
|  | TIFFSetField | set a tag’s value in the current directory |
|  | TIFFSetWarningHandler | set warning handler function |
|  | TIFFStripSize | returns size of a strip |
|  | TIFFRawStripSize | returns the number of bytes in a raw strip |
|  | TIFFSwabShort | swap bytes of short |
|  | TIFFSwabLong | swap bytes of long |
|  | TIFFSwabArrayOfShort | swap bytes of an array of shorts |
|  | TIFFSwabArrayOfLong | swap bytes of an array of longs |
|  | TIFFTileRowSize | return size of a row in a tile |
|  | TIFFTileSize | return size of a tile |
|  | TIFFUnRegisterCODEC | unregisters the codec |
|  | TIFFVGetField | return tag value in current directory |
|  | TIFFVGetFieldDefaulted | return tag value in current directory |
|  | TIFFVSetField | set a tag’s value in the current directory |
|  | TIFFVStripSize | returns the number of bytes in a strip |
|  | TIFFWarning | library warning handler |
|  | TIFFWriteDirectory | write the current directory |
|  | TIFFWriteEncodedStrip | compress and write a strip of data |
|  | TIFFWriteEncodedTile | compress and write a tile of data |
|  | TIFFWriteRawStrip | write a raw strip of data |
|  | TIFFWriteRawTile | write a raw tile of data |
|  | TIFFWriteScanline | write a scanline of data |
|  | TIFFWriteTile | compress and write a tile of data |
|  | TIFFXYZToRGB | perform CIE XYZ to RGB conversion |
|  | TIFFYCbCrToRGBInit | initialize YCbCr to RGB conversion state |
|  | TIFFYCbCrtoRGB | perform YCbCr to RGB conversion |

|  |  |
| --- | --- |
|  | Auxiliary functions: |

|  |  |  |
| --- | --- | --- |
|  | \_TIFFfree | free memory buffer |
|  | \_TIFFmalloc | dynamically allocate memory buffer |
|  | \_TIFFmemcmp | compare contents of the memory buffers |
|  | \_TIFFmemcpy | copy contents of the one buffer to another |
|  | \_TIFFmemset | fill memory buffer with a constant byte |
|  | \_TIFFrealloc | dynamically reallocate memory buffer |

## TAG USAGE

|  |  |
| --- | --- |
|  | The table below lists the TIFF tags that are recognized and handled by the library. If no use is indicated in the table, then the library reads and writes the tag, but does not use it internally. Note that some tags are meaningful only when a particul |