**TIFF CHANGE INFORMATION**

**Current Version**: v4.0.9  
**Previous Version**: [v4.0.8](http://docs.google.com/v4.0.8.html)  
**Master FTP Site**:  [download.osgeo.org](ftp://download.osgeo.org/libtiff), directory pub/libtiff  
**Master HTTP Site #1**: <http://www.simplesystems.org/libtiff/>  
**Master HTTP Site #2**: <http://libtiff.maptools.org/>

This document describes the changes made to the software between the *previous* and *current* versions (see above). If you don't find something listed here, then it was not done in this timeframe, or it was not considered important enough to be mentioned. The following information is located here:

* [Major Changes](#gjdgxs)
* [Changes in the software configuration](#30j0zll)
* [Changes in libtiff](#1fob9te)
* [Changes in the tools](#3znysh7)
* [Changes in the contrib area](#2et92p0)

**MAJOR CHANGES:**

* None

**CHANGES IN THE SOFTWARE CONFIGURATION:**

* test/Makefile.am: Add some tests for tiff2bw.
* \* .appveyor.yml, .travis.yml, build/travis-ci: apply patches 0001-ci-Travis-script-improvements.patch and 0002-ci-Invoke-helper-script-via-shell.patch by Roger Leigh (sent to mailing list)
* .travis.yml, build/travis-ci: new files from 0001-ci-Add-Travis-support-for-Linux-builds-with-Autoconf.patch by Roger Leigh (sent to mailing list on 2017-06-08) This patch adds support for the Travis-CI service.
* .appveyor.yml: new file from 0002-ci-Add-AppVeyor-support.patch by Roger Leigh (sent to mailing list on 2017-06-08) This patch adds a .appveyor.yml file to the top-level. This allows one to opt in to having a branch built on Windows with Cygwin, MinGW and MSVC automatically when a branch is pushed to GitHub, GitLab, BitBucket or any other supported git hosting service.
* CMakeLists.txt, test/CMakeLists.txt, test/TiffTestCommon.cmake: apply patch 0001-cmake-Improve-Cygwin-and-MingGW-test-support.patch from Roger Leigh (sent to mailing list on 2017-06-08) This patch makes the CMake build system support running the tests with MinGW or Cygwin.
* test/tiffcp-lzw-compat.sh, test/images/quad-lzw-compat.tiff: new files to test old-style LZW decompression
* test/common.sh, Makefile.am, CMakeList.txt: updated with above
* test/Makefile.am: add missing reference to images/quad-lzw-compat.tiff to fix "make distcheck". Patch by Roger Leigh
* nmake.opt: support a DEBUG=1 option, so as to adjust OPTFLAGS and use /MDd runtime in debug mode.

**CHANGES IN LIBTIFF:**

* libtiff/tif\_color.c: TIFFYCbCrToRGBInit(): stricter clamping to avoid int32 overflow in TIFFYCbCrtoRGB(). Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=1844 Credit to OSS Fuzz
* libtiff/tif\_getimage.c: initYCbCrConversion(): stricter validation for refBlackWhite coefficients values. To avoid invalid float->int32 conversion (when refBlackWhite[0] == 2147483648.f) Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=1907 Credit to OSS Fuzz
* libtiff/tif\_dirinfo.c, tif\_dirread.c: add \_TIFFCheckFieldIsValidForCodec(), and use it in TIFFReadDirectory() so as to ignore fields whose tag is a codec-specified tag but this codec is not enabled. This avoids TIFFGetField() to behave differently depending on whether the codec is enabled or not, and thus can avoid stack based buffer overflows in a number of TIFF utilities such as tiffsplit, tiffcmp, thumbnail, etc. Patch derived from 0063-Handle-properly-CODEC-specific-tags.patch (http://bugzilla.maptools.org/show\_bug.cgi?id=2580) by Raphaël Hertzog. Fixes: http://bugzilla.maptools.org/show\_bug.cgi?id=2580 http://bugzilla.maptools.org/show\_bug.cgi?id=2693 http://bugzilla.maptools.org/show\_bug.cgi?id=2625 (CVE-2016-10095) http://bugzilla.maptools.org/show\_bug.cgi?id=2564 (CVE-2015-7554) http://bugzilla.maptools.org/show\_bug.cgi?id=2561 (CVE-2016-5318) http://bugzilla.maptools.org/show\_bug.cgi?id=2499 (CVE-2014-8128) http://bugzilla.maptools.org/show\_bug.cgi?id=2441 http://bugzilla.maptools.org/show\_bug.cgi?id=2433
* libtiff/tif\_swab.c: if DISABLE\_CHECK\_TIFFSWABMACROS is defined, do not do the #ifdef TIFFSwabXXX checks. Make it easier for GDAL to rename the symbols of its internal libtiff copy.
* libtiff/tif\_dirread.c: fix regression of libtiff 4.0.8 in ChopUpSingleUncompressedStrip() regarding update of newly single-strip uncompressed files whose bytecount is 0. Before the change of 2016-12-03, the condition bytecount==0 used to trigger an early exit/disabling of strip chop. Re-introduce that in update mode. Otherwise this cause later incorrect setting for the value of StripByCounts/StripOffsets. ( https://trac.osgeo.org/gdal/ticket/6924 )
* libtiff/tif\_dirread.c: TIFFFetchStripThing(): limit the number of items read in StripOffsets/StripByteCounts tags to the number of strips to avoid excessive memory allocation. Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=2215 Credit to OSS Fuzz
* libtiff/tif\_getimage.c: avoid many (harmless) unsigned int overflows.
* libtiff/tif\_fax3.c: avoid unsigned int overflow in Fax3Encode2DRow(). Could potentially be a bug with huge rows.
* libtiff/tif\_jpeg.c: avoid (harmless) unsigned int overflow on tiled images.
* libtiff/tif\_dirread.c: avoid unsigned int overflow in EstimateStripByteCounts() and BYTECOUNTLOOKSBAD when file is too short.
* libtiff/tif\_predict.c: decorate legitimate functions where unsigned int overflow occur with TIFF\_NOSANITIZE\_UNSIGNED\_INT\_OVERFLOW \* libtiff/tif\_dirread.c: avoid unsigned int overflow in EstimateStripByteCounts()
* libtiff/tiffiop.h: add TIFF\_NOSANITIZE\_UNSIGNED\_INT\_OVERFLOW macro to disable CLang warnings raised by -fsanitize=undefined,unsigned-integer-overflow
* libtiff/tif\_jpeg.c: add anti-denial of service measure to avoid excessive CPU consumption on progressive JPEGs with a huge number of scans. See http://www.libjpeg-turbo.org/pmwiki/uploads/About/TwoIssueswiththeJPEGStandard.pdf Note: only affects libtiff since 2014-12-29 where support of non-baseline JPEG was added.
* libtiff/tif\_jpeg.c: error out at decoding time if anticipated libjpeg memory allocation is above 100 MB. libjpeg in case of multiple scans, which is allowed even in baseline JPEG, if components are spread over several scans and not interleavedin a single one, needs to allocate memory (or backing store) for the whole strip/tile. See http://www.libjpeg-turbo.org/pmwiki/uploads/About/TwoIssueswiththeJPEGStandard.pdf This limitation may be overriden by setting the LIBTIFF\_ALLOW\_LARGE\_LIBJPEG\_MEM\_ALLOC environment variable, or recompiling libtiff with a custom value of TIFF\_LIBJPEG\_LARGEST\_MEM\_ALLOC macro.
* libtiff/tif\_jbig.c: fix memory leak in error code path of JBIGDecode() Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2706 Reported by team OWL337
* libtiff/tif\_dirread.c: in TIFFReadDirEntryFloat(), check that a double value can fit in a float before casting. Patch by Nicolas RUFF
* libtiff/tiffiop.h, libtiff/tif\_jpeg.c, libtiff/tif\_jpeg\_12.c, libtiff/tif\_read.c: make TIFFReadScanline() works in CHUNKY\_STRIP\_READ\_SUPPORT mode with JPEG stream with multiple scans. Also make configurable through a LIBTIFF\_JPEG\_MAX\_ALLOWED\_SCAN\_NUMBER environment variable the maximum number of scans allowed. Defaults to 100.
* libtiff/tif\_read.c: TIFFFillTile(): add limitation to the number of bytes read in case td\_stripbytecount[strip] is bigger than reasonable, so as to avoid excessive memory allocation (similarly to what was done for TIFFFileStrip() on 2017-05-10)
* libtiff/tif\_getimage.c: use \_TIFFReadEncodedStripAndAllocBuffer(). Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2708 and https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=2433 . Credit to OSS Fuzz
* libtiff/tif\_read.c, tiffiop.h: add a \_TIFFReadEncodedStripAndAllocBuffer() function, variant of TIFFReadEncodedStrip() that allocates the decoded buffer only after a first successful TIFFFillStrip(). This avoids excessive memory allocation on corrupted files.
* libtiff/tif\_dirwrite.c: in TIFFWriteDirectoryTagCheckedXXXX() functions associated with LONG8/SLONG8 data type, replace assertion that the file is BigTIFF, by a non-fatal error. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2712 Reported by team OWL337
* libtiff/tif\_read.c: TIFFStartTile(): set tif\_rawcc to tif\_rawdataloaded when it is set. Similarly to TIFFStartStrip(). This issue was revealed by the change of 2017-06-30 in TIFFFileTile(), limiting the number of bytes read. But it could probably have been hit too in CHUNKY\_STRIP\_READ\_SUPPORT mode previously ? Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=2454 Credit to OSS Fuzz
* libtiff/tif\_error.c, tif\_warning.c: correctly use va\_list when both an old-style and new-style warning/error handlers are installed. Patch by Paavo Helde (sent on the mailing list)
* libtiff/tif\_getimage.c: use \_TIFFReadTileAndAllocBuffer(). Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=2470 Credit to OSS Fuzz.
* libtiff/tif\_read.c, tiffiop.h: add a \_TIFFReadEncodedTileAndAllocBuffer() and \_TIFFReadTileAndAllocBuffer() variants of TIFFReadEncodedTile() and TIFFReadTile() that allocates the decoded buffer only after a first successful TIFFFillTile(). This avoids excessive memory allocation on corrupted files.
* libtiff/tif\_pixarlog.c: avoid excessive memory allocation on decoding when RowsPerStrip tag is not defined (and thus td\_rowsperstrip == UINT\_MAX) Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=2554 Credit to OSS Fuzz
* libtiff/tif\_lzw.c: fix 4.0.8 regression in the decoding of old-style LZW compressed files.
* libtiff/tif\_lzw.c: fix potential out-of-buffer read on 1-byte LZW strips. Crashing issue only on memory mapped files, where the strip offset is the last byte of the file, and the file size is a multiple of one page size on the CPU architecture (typically 4096). Credit to myself :-)
* libtiff/tif\_dir.c: avoid potential null pointer dereference in \_TIFFVGetField() on corrupted TIFFTAG\_NUMBEROFINKS tag instance. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2713
* tools/tiff2pdf.c: prevent heap buffer overflow write in "Raw" mode on PlanarConfig=Contig input images. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2715 Reported by team OWL337
* libtiff/tif\_read.c: TIFFFillStrip() / TIFFFillTile(). Complementary fix for http://bugzilla.maptools.org/show\_bug.cgi?id=2708 in the isMapped() case, so as to avoid excessive memory allocation when we need a temporary buffer but the file is truncated.
* libtiff/tif\_read.c: TIFFFillStrip() / TIFFFillTile(). Complementary fix for http://bugzilla.maptools.org/show\_bug.cgi?id=2708 in the isMapped() case, so as to avoid excessive memory allocation when we need a temporary buffer but the file is truncated.
* libtiff/tif\_read.c: in TIFFFetchStripThing(), only grow the arrays that hold StripOffsets/StripByteCounts, when they are smaller than the expected number of striles, up to 1 million striles, and error out beyond. Can be tweaked by setting the environment variable LIBTIFF\_STRILE\_ARRAY\_MAX\_RESIZE\_COUNT. This partially goes against a change added on 2002-12-17 to accept those arrays of wrong sizes, but is needed to avoid denial of services. Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=2350 Credit to OSS Fuzz
* libtiff/tif\_read.c: in TIFFFetchStripThing(), only grow the arrays that hold StripOffsets/StripByteCounts, when they are smaller than the expected number of striles, up to 1 million striles, and error out beyond. Can be tweaked by setting the environment variable LIBTIFF\_STRILE\_ARRAY\_MAX\_RESIZE\_COUNT. This partially goes against a change added on 2002-12-17 to accept those arrays of wrong sizes, but is needed to avoid denial of services. Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=2350 Credit to OSS Fuzz
* libtiff/tif\_read.c: add protection against excessive memory allocation attempts in TIFFReadDirEntryArray() on short files. Effective for mmap'ed case. And non-mmap'ed case, but restricted to 64bit builds. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2675
* libtiff/tif\_read.c: add protection against excessive memory allocation attempts in TIFFReadDirEntryArray() on short files. Effective for mmap'ed case. And non-mmap'ed case, but restricted to 64bit builds. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2675
* libtiff/tif\_luv.c: LogLuvInitState(): avoid excessive memory allocation when RowsPerStrip tag is missing. Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=2683 Credit to OSS-Fuzz
* libtiff/tif\_getimage.c: gtTileContig() and gtTileSeparate(): properly break from loops on error when stoponerr is set, instead of going on iterating on row based loop.
* libtiff/tif\_getimage.c: fix fromskew computation when to-be-skipped pixel number is not a multiple of the horizontal subsampling, and also in some other cases. Impact putcontig8bitYCbCr44tile, putcontig8bitYCbCr42tile, putcontig8bitYCbCr41tile, putcontig8bitYCbCr21tile and putcontig8bitYCbCr12tile Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2637 (discovered by Agostino Sarubbo) and https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=2691 (credit to OSS Fuzz)
* libtiff/tif\_luv.c: further reduce memory requirements for temporary buffer when RowsPerStrip >= image\_length in LogLuvInitState() and LogL16InitState(). Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=2700 Credit to OSS Fuzz
* libtiff/tif\_dirwrite.c: replace assertion related to not finding the SubIFD tag by runtime check (in TIFFWriteDirectorySec()) Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2727 Reported by team OWL337
* libtiff/tif\_dirwrite.c: replace assertion to tag value not fitting on uint32 when selecting the value of SubIFD tag by runtime check (in TIFFWriteDirectoryTagSubifd()). Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2728 Reported by team OWL337
* libtiff/tif\_jpeg.c: accept reading the last strip of a JPEG compressed file if the codestream height is larger than the truncated height of the strip. Emit a warning in this situation since this is non compliant.
* libtiff/tiffiop.h, tif\_aux.c: redirect SeekOK() macro to a \_TIFFSeekoK() function that checks if the offset is not bigger than INT64\_MAX, so as to avoid a -1 error return code of TIFFSeekFile() to match a required seek to UINT64\_MAX/-1. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2726 Adapted from proposal by Nicolas Ruff.
* libtiff/tif\_dirread.c: add NULL check to avoid likely false positive null-pointer dereference warning by CLang Static Analyzer.
* libtiff/libtiff.def: add TIFFReadRGBAStripExt and TIFFReadRGBATileExt Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2735
* libtiff/tif\_jpeg.c: add compatibility with libjpeg-turbo 1.5.2 that honours max\_memory\_to\_use > 0. Cf https://github.com/libjpeg-turbo/libjpeg-turbo/issues/162
* libtiff/tif\_getimage.c: avoid floating point division by zero in initCIELabConversion() Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=3733 Credit to OSS Fuzz

**CHANGES IN THE TOOLS:**

* tools/tiff2pdf.c: prevent heap buffer overflow write in "Raw" mode on PlanarConfig=Contig input images. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2715 Reported by team OWL337
* tools/tiffset.c: fix setting a single value for the ExtraSamples tag (and other tags with variable number of values). So 'tiffset -s ExtraSamples 1 X'. This only worked when setting 2 or more values, but not just one.
* tools/fax2tiff.c (\_FAX\_Client\_Data): Pass FAX\_Client\_Data as the client data. This client data is not used at all at the moment, but it makes the most sense. Issue that the value of client\_data.fd was passed where a pointer is expected was reported via email by Gerald Schade on Sun, 29 Oct 2017.
* tools/tiff2pdf.c (t2p\_sample\_realize\_palette): Fix possible arithmetic overflow in bounds checking code and eliminate comparison between signed and unsigned type.
* tools/tiff2bw.c (main): Free memory allocated in the tiff2bw program. This is in response to the report associated with CVE-2017-16232 but does not solve the extremely high memory usage with the associated POC file.

**CHANGES IN THE CONTRIB AREA:**

* None

Last updated $Date: 2017-11-18 19:38:06 $.