**TIFF CHANGE INFORMATION**

**Current Version**: v4.0.8  
**Previous Version**: [v4.0.7](http://docs.google.com/v4.0.7.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:**

* None

**CHANGES IN LIBTIFF:**

* libtiff/tif\_getimage.c, libtiff/tif\_open.c: add parenthesis to fix cppcheck clarifyCalculation warnings \* libtiff/tif\_predict.c, libtiff/tif\_print.c: fix printf unsigned vs signed formatting (cppcheck invalidPrintfArgType\_uint warnings)
* libtiff/tif\_read.c, libtiff/tiffiop.h: fix uint32 overflow in TIFFReadEncodedStrip() that caused an integer division by zero. Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2596
* libtiff/tif\_pixarlog.c, libtiff/tif\_luv.c: fix heap-based buffer overflow on generation of PixarLog / LUV compressed files, with ColorMap, TransferFunction attached and nasty plays with bitspersample. The fix for LUV has not been tested, but suffers from the same kind of issue of PixarLog. Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2604
* libtiff/tif\_strip.c: revert the change in TIFFNumberOfStrips() done for http://bugzilla.maptools.org/show\_bug.cgi?id=2587 / CVE-2016-9273 since the above change is a better fix that makes it unnecessary.
* libtiff/tif\_dirread.c: modify ChopUpSingleUncompressedStrip() to instanciate compute ntrips as TIFFhowmany\_32(td->td\_imagelength, rowsperstrip), instead of a logic based on the total size of data. Which is faulty is the total size of data is not sufficient to fill the whole image, and thus results in reading outside of the StripByCounts/StripOffsets arrays when using TIFFReadScanline(). Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2608.
* libtiff/tif\_ojpeg.c: make OJPEGDecode() early exit in case of failure in OJPEGPreDecode(). This will avoid a divide by zero, and potential other issues. Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2611
* libtiff/tif\_write.c: fix misleading indentation as warned by GCC.
* libtiff/tif\_fax3.h: revert change done on 2016-01-09 that made Param member of TIFFFaxTabEnt structure a uint16 to reduce size of the binary. It happens that the Hylafax software uses the tables that follow this typedef (TIFFFaxMainTable, TIFFFaxWhiteTable, TIFFFaxBlackTable), although they are not in a public libtiff header. Raised by Lee Howard. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2636
* libtiff/tiffio.h, libtiff/tif\_getimage.c: add TIFFReadRGBAStripExt() and TIFFReadRGBATileExt() variants of the functions without ext, with an extra argument to control the stop\_on\_error behaviour.
* libtiff/tif\_getimage.c: fix potential memory leaks in error code path of TIFFRGBAImageBegin(). Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2627
* libtiff/tif\_jpeg.c: increase libjpeg max memory usable to 10 MB instead of libjpeg 1MB default. This helps when creating files with "big" tile, without using libjpeg temporary files. Related to https://trac.osgeo.org/gdal/ticket/6757
* libtiff/tif\_jpeg.c: avoid integer division by zero in JPEGSetupEncode() when horizontal or vertical sampling is set to 0. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2653
* libtiff/tif\_dirwrite.c: in TIFFWriteDirectoryTagCheckedRational, replace assertion by runtime check to error out if passed value is strictly negative. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2535
* libtiff/tif\_dirread.c: avoid division by floating point 0 in TIFFReadDirEntryCheckedRational() and TIFFReadDirEntryCheckedSrational(), and return 0 in that case (instead of infinity as before presumably) Apparently some sanitizers do not like those divisions by zero. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2644
* libtiff/tif\_dir.c, tif\_dirread.c, tif\_dirwrite.c: implement various clampings of double to other data types to avoid undefined behaviour if the output range isn't big enough to hold the input value. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2643 http://bugzilla.maptools.org/show\_bug.cgi?id=2642 http://bugzilla.maptools.org/show\_bug.cgi?id=2646 http://bugzilla.maptools.org/show\_bug.cgi?id=2647
* libtiff/tif\_jpeg.c: validate BitsPerSample in JPEGSetupEncode() to avoid undefined behaviour caused by invalid shift exponent. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2648
* libtiff/tif\_read.c: avoid potential undefined behaviour on signed integer addition in TIFFReadRawStrip1() in isMapped() case. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2650
* libtiff/tif\_getimage.c: add explicit uint32 cast in putagreytile to avoid UndefinedBehaviorSanitizer warning. Patch by Nicolás Peña. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2658
* libtiff/tif\_read.c: TIFFReadBufferSetup(): use \_TIFFcalloc() to zero initialize tif\_rawdata. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2651
* libtiff/tiffio.h, tif\_unix.c, tif\_win32.c, tif\_vms.c: add \_TIFFcalloc()
* libtiff/tif\_luv.c, tif\_lzw.c, tif\_packbits.c: return 0 in Encode functions instead of -1 when TIFFFlushData1() fails. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2130
* libtiff/tif\_ojpeg.c: fix leak in OJPEGReadHeaderInfoSecTablesQTable, OJPEGReadHeaderInfoSecTablesDcTable and OJPEGReadHeaderInfoSecTablesAcTable when read fails. Patch by Nicolás Peña. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2659
* libtiff/tif\_jpeg.c: only run JPEGFixupTagsSubsampling() if the YCbCrSubsampling tag is not explicitly present. This helps a bit to reduce the I/O amount when the tag is present (especially on cloud hosted files).
* libtiff/tif\_lzw.c: in LZWPostEncode(), increase, if necessary, the code bit-width after flushing the remaining code and before emitting the EOI code. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=1982
* libtiff/tif\_pixarlog.c: fix memory leak in error code path of PixarLogSetupDecode(). Patch by Nicolás Peña. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2665
* libtiff/tif\_fax3.c, tif\_predict.c, tif\_getimage.c: fix GCC 7 -Wimplicit-fallthrough warnings.
* libtiff/tif\_dirread.c: fix memory leak in non DEFER\_STRILE\_LOAD mode (ie default) when there is both a StripOffsets and TileOffsets tag, or a StripByteCounts and TileByteCounts Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2689
* libtiff/tif\_ojpeg.c: fix potential memory leak in OJPEGReadHeaderInfoSecTablesQTable, OJPEGReadHeaderInfoSecTablesDcTable and OJPEGReadHeaderInfoSecTablesAcTable Patch by Nicolás Peña. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2670
* libtiff/tif\_fax3.c: avoid crash in Fax3Close() on empty file. Patch by Alan Coopersmith + complement by myself. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2673
* libtiff/tif\_read.c: TIFFFillStrip(): add limitation to the number of bytes read in case td\_stripbytecount[strip] is bigger than reasonable, so as to avoid excessive memory allocation.
* libtiff/tif\_zip.c, tif\_pixarlog.c, tif\_predict.c: fix memory leak when the underlying codec (ZIP, PixarLog) succeeds its setupdecode() method, but PredictorSetup fails. Credit to OSS-Fuzz (locally run, on GDAL)
* libtiff/tif\_read.c: TIFFFillStrip() and TIFFFillTile(): avoid excessive memory allocation in case of shorten files. Only effective on 64 bit builds and non-mapped cases. Credit to OSS-Fuzz (locally run, on GDAL)
* libtiff/tif\_read.c: TIFFFillStripPartial() / TIFFSeek(), avoid potential integer overflows with read\_ahead in CHUNKY\_STRIP\_READ\_SUPPORT mode. Should especially occur on 32 bit platforms.
* libtiff/tif\_read.c: TIFFFillStripPartial(): avoid excessive memory allocation in case of shorten files. Only effective on 64 bit builds. Credit to OSS-Fuzz (locally run, on GDAL)
* libtiff/tif\_read.c: update tif\_rawcc in CHUNKY\_STRIP\_READ\_SUPPORT mode with tif\_rawdataloaded when calling TIFFStartStrip() or TIFFFillStripPartial(). This avoids reading beyond tif\_rawdata when bytecount > tif\_rawdatasize. Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=1545. Credit to OSS-Fuzz
* libtiff/tif\_color.c: avoid potential int32 overflow in TIFFYCbCrToRGBInit() Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=1533 Credit to OSS-Fuzz
* libtiff/tif\_pixarlog.c, tif\_luv.c: avoid potential int32 overflows in multiply\_ms() and add\_ms(). Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=1558 Credit to OSS-Fuzz
* libtiff/tif\_packbits.c: fix out-of-buffer read in PackBitsDecode() Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=1563 Credit to OSS-Fuzz
* libtiff/tif\_luv.c: LogL16InitState(): avoid excessive memory allocation when RowsPerStrip tag is missing. Credit to OSS-Fuzz (locally run, on GDAL)
* libtiff/tif\_lzw.c: update dec\_bitsleft at beginning of LZWDecode(), and update tif\_rawcc at end of LZWDecode(). This is needed to properly work with the latest chnges in tif\_read.c in CHUNKY\_STRIP\_READ\_SUPPORT mode.
* libtiff/tif\_pixarlog.c: PixarLogDecode(): resync tif\_rawcp with next\_in and tif\_rawcc with avail\_in at beginning and end of function, similarly to what is done in LZWDecode(). Likely needed so that it works properly with latest chnges in tif\_read.c in CHUNKY\_STRIP\_READ\_SUPPORT mode. But untested...
* libtiff/tif\_getimage.c: initYCbCrConversion(): add basic validation of luma and refBlackWhite coefficients (just check they are not NaN for now), to avoid potential float to int overflows. Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=1663 Credit to OSS Fuzz
* libtiff/tif\_read.c: \_TIFFVSetField(): fix outside range cast of double to float. Credit to Google Autofuzz project
* libtiff/tif\_getimage.c: initYCbCrConversion(): check luma[1] is not zero to avoid division by zero. Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=1665 Credit to OSS Fuzz
* libtiff/tif\_read.c: \_TIFFVSetField(): fix outside range cast of double to float. Credit to Google Autofuzz project
* libtiff/tif\_getimage.c: initYCbCrConversion(): check luma[1] is not zero to avoid division by zero. Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=1665 Credit to OSS Fuzz
* libtiff/tif\_getimage.c: initYCbCrConversion(): stricter validation for refBlackWhite coefficients values. To avoid invalid float->int32 conversion. Fixes https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=1718 Credit to OSS Fuzz

**CHANGES IN THE TOOLS:**

* tools/fax2tiff.c (main): Applied patch by Jörg Ahrens to fix passing client data for Win32 builds using tif\_win32.c (USE\_WIN32\_FILEIO defined) for file I/O. Patch was provided via email on November 20, 2016.
* tools/tiffcp.c: avoid uint32 underflow in cpDecodedStrips that can cause various issues, such as buffer overflows in the library. Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2598
* tools/tiffcrop.c: fix readContigStripsIntoBuffer() in -i (ignore) mode so that the output buffer is correctly incremented to avoid write outside bounds. Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2620
* tools/tiffcrop.c: add 3 extra bytes at end of strip buffer in readSeparateStripsIntoBuffer() to avoid read outside of heap allocated buffer. Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2621
* tools/tiffcrop.c: fix integer division by zero when BitsPerSample is missing. Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2619
* tools/tiffinfo.c: fix null pointer dereference in -r mode when the image has no StripByteCount tag. Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2594
* tools/tiffcp.c: avoid potential division by zero is BitsPerSamples tag is missing. Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2597
* tools/tif\_dir.c: when TIFFGetField(, TIFFTAG\_NUMBEROFINKS, ) is called, limit the return number of inks to SamplesPerPixel, so that code that parses ink names doesn't go past the end of the buffer. Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2599
* tools/tiffcp.c: avoid potential division by zero is BitsPerSamples tag is missing. Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2607
* tools/tiffcp.c: fix uint32 underflow/overflow that can cause heap-based buffer overflow. Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2610
* tools/tiffcp.c: replace assert( (bps % 8) == 0 ) by a non assert check. Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2605
* tools/tiff2ps.c: fix 2 heap-based buffer overflows (in PSDataBW and PSDataColorContig). Reported by Agostino Sarubbo. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2633 and http://bugzilla.maptools.org/show\_bug.cgi?id=2634.
* tools/tiff2pdf.c: prevent heap-based buffer overflow in -j mode on a paletted image. Note: this fix errors out before the overflow happens. There could probably be a better fix. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2635
* tools/tiff2pdf.c: fix wrong usage of memcpy() that can trigger unspecified behaviour. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2638
* tools/tiff2pdf.c: avoid potential invalid memory read in t2p\_writeproc. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2639
* tools/tiff2pdf.c: avoid potential heap-based overflow in t2p\_readwrite\_pdf\_image\_tile(). Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2640
* tools/tiffcrop.c: remove extraneous TIFFClose() in error code path, that caused double free. Related to http://bugzilla.maptools.org/show\_bug.cgi?id=2535
* tools/tiffcp.c: error out cleanly in cpContig2SeparateByRow and cpSeparate2ContigByRow if BitsPerSample != 8 to avoid heap based overflow. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2656 and http://bugzilla.maptools.org/show\_bug.cgi?id=2657
* tools/raw2tiff.c: avoid integer division by zero. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2631
* tools/tiff2ps.c: call TIFFClose() in error code paths.
* tools/fax2tiff.c: emit appropriate message if the input file is empty. Patch by Alan Coopersmith. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2672
* tools/tiff2bw.c: close TIFF handle in error code path. Fixes http://bugzilla.maptools.org/show\_bug.cgi?id=2677

**CHANGES IN THE CONTRIB AREA:**

* None

Last updated $Date: 2017-05-21 17:47:46 $.