Exiv2 - Bug #769

"Assertion `sv == d' failed" in 0.21.1 (r2474)

24 Apr 2011 16:55 - Derek Chen-Becker

Status:	Closed	**Start date:**	24 Apr 2011
Priority:	Normal	**Due date:**	
Assignee:	Andreas Huggel	**% Done:**	100%
Category:	tiff parser	**Estimated time:**	0.00 hour
Target version:	0.22		

Description

I thought that this was #752 but I pulled and build from SVN/trunk (r2474) and I'm still getting a crash in gThumb when rotating an image (same image every time). Full stack trace:

```
#0  0x00007fd985f476dd in waitpid () from /lib/libpthread.so.0
#1  0x00007fd98645ed11 in g_spawn_sync (working_directory=<value optimized out>, argv=<value optim
ized out>, envp=<value optimized out>, flags=<value optimized out>, child_setup=<value optimized o
ut>, user_data=<value optimized out>, standard_output=0x0, standard_error=0x0, exit_status=0x0, er
ror=0x7fffae83c038) at /build/buildd/glib2.0-2.26.1/glib/gspawn.c:392
#2  0x00007fd98645f029 in g_spawn_command_line_sync (command_line=<value optimized out>, standard_
output=0x0, standard_error=0x0, exit_status=0x0, error=0x7fffae83c038) at /build/buildd/glib2.0-2.
26.1/glib/gspawn.c:706
#3  0x00007fd98036f2c4 in ?? () from /usr/lib/gtk-2.0/modules/libgnomesegvhandler.so
#4  <signal handler called>
#5  0x00007fd985be8ba5 in raise () from /lib/libc.so.6
#6  0x00007fd985bec6b0 in abort () from /lib/libc.so.6
#7  0x00007fd985be1a71 in __assert_fail () from /lib/libc.so.6
#8  0x00007fd989dbb4d8 in Exiv2::Internal::TiffDirectory::doWrite (this=0x1b5b540, ioWrapper=...,
byteOrder=Exiv2::littleEndian, offset=704, valueIdx=330, dataIdx=2680, imageIdx=@0x7fffae83cfd8) a
t tiffcomposite.cpp:1156
#9  0x00007fd989dbaee9 in Exiv2::Internal::TiffComponent::write (this=0x1b5b540, ioWrapper=..., by
teOrder=Exiv2::littleEndian, offset=704, valueIdx=4294967295, dataIdx=4294967295, imageIdx=@0x7fff
ae83cfd8) at tiffcomposite.cpp:1065
#10 0x00007fd989dbc142 in Exiv2::Internal::TiffIfdMakernote::doWrite (this=0x1b5b520, ioWrapper=..
., byteOrder=Exiv2::littleEndian, offset=704, imageIdx=@0x7fffae83cfd8) at tiffcomposite.cpp:1362
#11 0x00007fd989dbaee9 in Exiv2::Internal::TiffComponent::write (this=0x1b5b520, ioWrapper=..., by
teOrder=Exiv2::littleEndian, offset=704, valueIdx=4294967295, dataIdx=4294967295, imageIdx=@0x7fff
ae83cfd8) at tiffcomposite.cpp:1065
#12 0x00007fd989dbc08a in Exiv2::Internal::TiffMnEntry::doWrite (this=0x1b59890, ioWrapper=..., by
teOrder=Exiv2::littleEndian, offset=210, valueIdx=494, dataIdx=3198, imageIdx=@0x7fffae83cfd8) at
tiffcomposite.cpp:1346
#13 0x00007fd989dbaee9 in Exiv2::Internal::TiffComponent::write (this=0x1b59890, ioWrapper=..., by
teOrder=Exiv2::littleEndian, offset=210, valueIdx=494, dataIdx=3198, imageIdx=@0x7fffae83cfd8) at
tiffcomposite.cpp:1065
#14 0x00007fd989dbb4ae in Exiv2::Internal::TiffDirectory::doWrite (this=0x1b57fe0, ioWrapper=...,
byteOrder=Exiv2::littleEndian, offset=210, valueIdx=494, dataIdx=3198, imageIdx=@0x7fffae83cfd8) a
t tiffcomposite.cpp:1155
#15 0x00007fd989dbaee9 in Exiv2::Internal::TiffComponent::write (this=0x1b57fe0, ioWrapper=..., by
teOrder=Exiv2::littleEndian, offset=210, valueIdx=4294967295, dataIdx=4294967295, imageIdx=@0x7fff
ae83cfd8) at tiffcomposite.cpp:1065
#16 0x00007fd989dbcc86 in Exiv2::Internal::TiffSubIfd::doWriteData (this=0x1b583d0, ioWrapper=...,
 byteOrder=Exiv2::littleEndian, offset=8, dataIdx=202, imageIdx=@0x7fffae83cfd8) at tiffcomposite.
cpp:1513
#17 0x00007fd989dbc96f in Exiv2::Internal::TiffComponent::writeData (this=0x1b583d0, ioWrapper=...
, byteOrder=Exiv2::littleEndian, offset=8, dataIdx=202, imageIdx=@0x7fffae83cfd8) at tiffcomposite
.cpp:1449
#18 0x00007fd989dbc9ec in Exiv2::Internal::TiffDirectory::doWriteData (this=0x1b57060, ioWrapper=.
.., byteOrder=Exiv2::littleEndian, offset=8, dataIdx=202, imageIdx=@0x7fffae83cfd8) at tiffcomposi
te.cpp:1460
#19 0x00007fd989dbc96f in Exiv2::Internal::TiffComponent::writeData (this=0x1b57060, ioWrapper=...
, byteOrder=Exiv2::littleEndian, offset=8, dataIdx=202, imageIdx=@0x7fffae83cfd8) at tiffcomposite
.cpp:1449
#20 0x00007fd989dbb5f0 in Exiv2::Internal::TiffDirectory::doWrite (this=0x1b57060, ioWrapper=...,
```

```
byteOrder=Exiv2::littleEndian, offset=8, valueIdx=202, dataIdx=202, imageIdx=@0x7fffae83cfd8) at t
iffcomposite.cpp:1172
#21 0x00007fd989dbaee9 in Exiv2::Internal::TiffComponent::write (this=0x1b57060, ioWrapper=..., by
teOrder=Exiv2::littleEndian, offset=8, valueIdx=4294967295, dataIdx=4294967295, imageIdx=@0x7fffae
83cfd8) at tiffcomposite.cpp:1065
#22 0x00007fd989dc670d in Exiv2::Internal::TiffParserWorker::encode (io=..., pData=0x1564850 "II*"
, size=13302, exifData=..., iptcData=..., xmpData=..., root=131072, findEncoderFct=0x7fd989dc5cd0
<Exiv2::Internal::TiffMapping::findEncoder(std::string const&, uint32_t, Exiv2::Internal::IfdId)>,
 pHeader=0x160f0e0) at tiffimage.cpp:1852
#23 0x00007fd989d3c066 in Exiv2::ExifParser::encode (blob=..., pData=0x1564850 "II*", size=13302,
byteOrder=Exiv2::littleEndian, exifData=...) at exif.cpp:723
#24 0x00007fd989d56e7c in Exiv2::JpegBase::doWriteMetadata (this=0x156baa0, outIo=...) at jpgimage
.cpp:674
#25 0x00007fd989d55a9b in Exiv2::JpegBase::writeMetadata (this=0x156baa0) at jpgimage.cpp:499
#26 0x00007fd97e6ea712 in exiv2_write_metadata_private (image=<value optimized out>, info=<value o
ptimized out>, pixbuf=<value optimized out>) at exiv2-utils.cpp:1027
#27 0x00007fd97e6eb23b in exiv2_write_metadata_to_buffer (buffer=0x7fffae83e968, buffer_size=0x7ff
fae83e960, info=0x134d100, pixbuf=0x0, error=<value optimized out>) at exiv2-utils.cpp:1091
#28 0x00007fd97e6efe0c in exiv2_jpeg_tran_cb (tran_info=0x7fffae83e130) at main.c:212
#29 0x00007fd98640aeef in g_hook_list_marshal (hook_list=0x11bae20, may_recurse=1, marshaller=0x46
6440 <invoke_marshaller_1>, data=0x1525130) at /build/buildd/glib2.0-2.26.1/glib/ghook.c:387
#30 0x000000000046680e in gth_hook_invoke (name=<value optimized out>, first_data=<value optimized
 out>) at gth-hook.c:269
#31 0x00007fd97ed0ccb6 in jpegtran (in_buffer=<value optimized out>, in_buffer_size=<value optimiz
ed out>, out_buffer=<value optimized out>, out_buffer_size=<value optimized out>, transformation=<
value optimized out>, mcu_action=<value optimized out>, error=0x7fffae83e958) at jpegtran.c:337
#32 0x00007fd97ef14109 in file_buffer_ready_cb (buffer=0x1655d18, count=1308364, error=0x0, user_d
ata=<value optimized out>) at rotation-utils.c:289
#33 0x000000000043c3ea in load_file__stream_read_cb (source_object=<value optimized out>, result=<
value optimized out>, user_data=<value optimized out>) at gio-utils.c:2011
#34 0x00007fd988a5a2e9 in async_ready_callback_wrapper (source_object=0x15d7920, res=0x16319e0, us
er_data=0x1654cf0) at /build/buildd/glib2.0-2.26.1/gio/ginputstream.c:470
#35 0x00007fd988a69f28 in complete_in_idle_cb_for_thread (_data=<value optimized out>) at /build/b
uildd/glib2.0-2.26.1/gio/gsimpleasyncresult.c:757
#36 0x00007fd986418342 in g_main_dispatch (context=0x1136510) at /build/buildd/glib2.0-2.26.1/glib
/gmain.c:2149
#37 g_main_context_dispatch (context=0x1136510) at /build/buildd/glib2.0-2.26.1/glib/gmain.c:2702
#38 0x00007fd98641c2a8 in g_main_context_iterate (context=0x1136510, block=<value optimized out>,
dispatch=<value optimized out>, self=<value optimized out>) at /build/buildd/glib2.0-2.26.1/glib/g
main.c:2780
#39 0x00007fd98641c7b5 in g_main_loop_run (loop=0x10fbd70) at /build/buildd/glib2.0-2.26.1/glib/gm
ain.c:2988
#40 0x00007fd9893193e7 in IA__gtk_main () at /build/buildd/gtk+2.0-2.22.0/gtk/gtkmain.c:1237
#41 0x00000000004911bd in main (argc=1, argv=0x7fffae83ed28) at main.c:449
```

The image is attached. This is happening a lot with images taken with my new camera (Canon S95), so I'm not sure if there's some makernote field that's not getting parsed properly or if something else is really broken.

Thanks,

Derek

Associated revisions

Revision 2562 - 20 Jul 2011 22:01 - Andreas Huggel

#769: Fixed calculation of binary array size in the case when the array elements are not sorted in ascending order, added test case.

History

#1 - 25 Apr 2011 18:06 - Andreas Huggel

The image looks ok, I can't see anything wrong with its metadata. Definitely not the same problem as #752.

I can't reproduce the issue with the exiv2 command line tool. Adding a tag works fine. The backtrace shows that exiv2 is serializing the metadata when it crashes, maybe related to the actual data that was changed. What tags does gThumb modify here?

(cc Mike from gThumb)

#2 - 26 Apr 2011 17:12 - Derek Chen-Becker

I'll have to confirm but at the very least it's going to modify the orientation tags and probably the pixel X/Y dimensions. It also appears to remove the EXIF thumbnail from my images (thank goodness, I wish I could just turn that off in my camera). gThumb adds a "Software" tag equal to "gthumb <version". Let me compare EXIF tags from the image in question and one of the images that gthumb successfully rotates.

#3 - 26 Apr 2011 17:19 - Derek Chen-Becker

- File IMG_2280.exif added

- File IMG_2281.exif added

OK, here are the tags I can get from "exiv2 -P E pr <file>" for an image that doesn't crash (IMG_2280.JPG) and the one that does (IMG_2281.JPG). Besides the obvious differences in shooting conditions, it looks like a few of the tags get removed (Exif.Canon.OwnerName, Exif.Image.ImageDescription, etc). If you would like more detailed EXIF data just let me know the command line I should use.

#4 - 26 Apr 2011 17:44 - Andreas Huggel

Thanks. Since they are not too large, can you just attach a "before" and "after" picture?

#5 - 26 Apr 2011 17:51 - Andreas Huggel

This is what I'll do with the information, but probably not before the weekend. If you have time, you can try yourself: Create an exiv2 command script that transforms the metadata of the "before" picture to the "after" state (-m option, man exiv2). Then apply it to the picture that crashes. Does it crash too, i.e., can you reproduce the crash with these exiv2 commands?

#6 - 26 Apr 2011 19:34 - Derek Chen-Becker

Andreas Huggel wrote:

> This is what I'll do with the information, but probably not before the weekend. If you have time, you can try yourself: Create an exiv2 command script that transforms the metadata of the "before" picture to the "after" state (-m option, man exiv2). Then apply it to the picture that crashes. Does it crash too, i.e., can you reproduce the crash with these exiv2 commands?

I'll have time to work on this on Sunday. Let me see what I find. I'll take a test photo to do before and after and see if I can make a script that replicates the changes.

Thanks!

#7 - 16 May 2011 09:05 - Derek Chen-Becker

Well, I've tried replicating using a mod script, but I can't figure out how to modify things like MakerNotes (which is changed). Let me take a test shot and upload before and after images (assuming I can get an image that won't crash gThumb)

#8 - 16 May 2011 09:12 - Derek Chen-Becker

- File IMG_2801-orig.JPG added

- File IMG_2801.JPG added

OK, got lucky. The attached images are the before (-orig) and after images with a gThumb rotation.

#9 - 19 May 2011 09:16 - Andreas Huggel

- File 2801.txt added

- Status changed from New to Feedback

I can't reproduce the issue. The attach command file transforms the metadata of the "before" image to the "after" state but also applies without any crash on IMG_2281.JPG.

There is definitely something wrong here, but I can't help without being able to reproduce the crash.

Andreas

#10 - 19 May 2011 13:27 - Derek Chen-Becker

Understood. The command file I sent isn't 100% correct, FYI. For one thing, I don't transform the maker notes or a few other things that I wasn't sure how to change. In any case I'm working around the issue now by using exiftran from the command line.

#11 - 19 May 2011 19:52 - Andreas Huggel

Makernote tags can be modified like any other tags. The only special cases here are tags with offsets, which are computed during writing. You don't need to set them. See the commands in the attached file 2801.txt. After applying these commands to IMG_2801-orig.JPG, exiv2 reports the same

metadata in IMG_2801-orig.JPG as in IMG_2801.JPG, except for a difference in the sequence of the makernote tags.

Andreas

#12 - 19 May 2011 22:44 - Derek Chen-Becker

Ah, sorry, I misunderstood. I'll bump this up the the gThumb folks and see if they can look at it. Just looking at their source for the image rotation extension, it looks like they're wiping all EXIF data from the image and then rebuilding it, rather than modifying one or two attributes.

#13 - 17 Jul 2011 08:10 - Paolo Bacchilega

Andreas Huggel wrote:

> Makernote tags can be modified like any other tags. The only special cases here are tags with offsets, which are computed during writing. You don't need to set them.

Is there a way to know if a tag has an offset or, in general, if it's safe to set it?

Paolo (gThumb developer)

#14 - 17 Jul 2011 20:50 - Derek Chen-Becker

FYI, since Paolo joined in I wanted to mention that I'm still affected by this. I'm working around it currently by disabling auto-rotate on import and then I have my own script (using exiftran) that deals with the images :(

#15 - 17 Jul 2011 23:22 - Andreas Huggel

Paolo Bacchilega wrote:

> Andreas Huggel wrote:
>
> > Makernote tags can be modified like any other tags. The only special cases here are tags with offsets, which are computed during writing. You don't need to set them.
>
> Is there a way to know if a tag has an offset or, in general, if it's safe to set it?
>
> Paolo (gThumb developer)

Setting these offset tags doesn't affect the way exiv2 works and doesn't result in a corrupted image. The value you set is ignored and instead exiv2 uses the calculated offset. If the tag is not needed it will not be written at all to ensure the TIFF structure remains intact and prevent a dangling pointer.

There is no simple way to recognize these tags. The most common ones are:

```
Exif.Image.ExifTag              : Offset of the Exif IFD
Exif.Photo.InteroperabilityTag : Offset of the Interoperability IFD
Exif.Image.GPSTag               : Offset of the GPS IFD
```

You'd have to read specifications or look in the Exiv2 source code for others (look for newTiffSubIfd in tiffimage.cpp).

Andreas

#16 - 18 Jul 2011 01:02 - Paolo Bacchilega

> Setting these offset tags doesn't affect the way exiv2 works and doesn't result in a corrupted image.

I still don't understand how to avoid the sv==d assertion failure.

I'll try to describe how gThumb uses exiv2.

Metadata is read in the exiv2_read_metadata function (http://git.gnome.org/browse/gthumb/tree/extensions/exiv2_tools/exiv2-utils.cpp#n557), for each metadatum the key, description, formatted value, raw value and type name are stored in the GthMetadata structure.

Metadata is written in the exiv2_write_metadata_private function (http://git.gnome.org/browse/gthumb/tree/extensions/exiv2_tools/exiv2-utils.cpp#n861) where values are created using the command:

Exiv2::Value::AutoPtr value = Exiv2::Value::create (Exiv2::TypeInfo::typeId (value_type));
value->read (raw_value);

where value_type is the string returned by md->typeName() and raw_value by md->value() in the exiv2_read_metadata function.

Do you see any problem with this way of saving metadata ?

#17 - 18 Jul 2011 02:48 - Andreas Huggel

> I still don't understand how to avoid the sv==d assertion failure.

That's probably a bug. I'd like to understand what triggers it. If you can reproduce the problem, can you dump the contents of the ExifData container after reading the metadata and again, before writing (from the newly constructed ExifData container)? Something like this http://www.exiv2.org/doc/exifprint_8cpp-example.html#_a1 can be used to print the data.

> Do you see any problem with this way of saving metadata ?

It differs from the common use-case, but I can't immediately tell why it sometimes doesn't work. The usual case is that applications keep the ExifData container and modify it as needed (add/mod/del tags). In gthumb you take the tags out of that into another list and later build a new ExifData object.

Andreas

#18 - 18 Jul 2011 07:13 - Paolo Bacchilega

- File exif-dump-before.txt added

- File exif-data-after.txt added

- File exif-data-after-no-crash.txt added

Andreas Huggel wrote:

> > I still don't understand how to avoid the sv==d assertion failure.
>
> That's probably a bug. I'd like to understand what triggers it. If you can reproduce the problem, can you dump the contents of the ExifData container after reading the metadata and again, before writing (from the newly constructed ExifData container)?

I can reproduce the problem when changing a set of 600 images, the s==d assertion is triggered always by the same image, however if I try to change only that image, or a smaller set of images containing that same image, there is no error.

Explanation of the attached files:

exif-dump-before.txt => this is the ExifData of the original image
exif-dump-before-after.txt => this is the ExifData before saving the image and when the image is part of a 600 images set
exif-dump-before-after-no-crash.txt => this is the ExifData before saving the image and when the image is the only file to be changed (in this case there is no crash)

#19 - 19 Jul 2011 04:26 - Paolo Bacchilega

- File exif-data-before-after-diff.txt added

Here is the diff between exif-dump-before.txt and exif-data-after.txt, maybe this way it's easier to spot the problem.

#20 - 19 Jul 2011 21:03 - Andreas Huggel

- Category set to tiff parser

- Status changed from Feedback to Assigned

- Assignee set to Andreas Huggel

Thanks Paolo! With this data I can finally reproduce the assertion here!

#21 - 19 Jul 2011 21:06 - Andreas Huggel

- File exiv2-bug769.jpg added

- File exif-data-after.txt.cmd added

$ exiv2 -m exif-data-after.txt.cmd exiv2-bug769.jpg
Warning: Directory Thumbnail, entry 0x0201: Data area exceeds data buffer, ignoring it.
Warning: Directory Thumbnail, entry 0x0201: Data area exceeds data buffer, ignoring it.

exiv2: tiffcomposite.cpp:1156: virtual uint32_t Exiv2::Internal::TiffDirectory::doWrite(Exiv2::Internal::IoWrapper&, Exiv2::ByteOrder, int32_t, uint32_t, uint32_t, uint32_t&): Assertion `sv == d' failed.
Aborted

#22 - 20 Jul 2011 22:12 - Andreas Huggel

- Status changed from Assigned to Resolved

- Target version set to 0.22

- % Done changed from 0 to 100

The bug that I can reproduce here shows if one tries to add binary array elements out of sequence. That happens in the above cases because Exif.CanonCs.FlashDetails is added at the end of the list.

r2562 fixes this issue. Please test and confirm that all assertions are gone with the current trunk. Alternatively, the changes should also apply as a binary compatible patch to 0.21.1.

Andreas

#23 - 21 Jul 2011 01:07 - Paolo Bacchilega

The patch fixes the bug for me as well, thank you.

Paolo

#24 - 18 Sep 2011 05:27 - Andreas Huggel

- Status changed from Resolved to Closed

Files

IMG_2281.JPG	1.25 MB	24 Apr 2011	Derek Chen-Becker
IMG_2280.exif	7.79 KB	26 Apr 2011	Derek Chen-Becker
IMG_2281.exif	8.38 KB	26 Apr 2011	Derek Chen-Becker
IMG_2801-orig.JPG	450 KB	16 May 2011	Derek Chen-Becker
IMG_2801.JPG	441 KB	16 May 2011	Derek Chen-Becker
2801.txt	532 Bytes	19 May 2011	Andreas Huggel
exif-dump-before.txt	13.3 KB	18 Jul 2011	Paolo Bacchilega
exif-data-after.txt	12.7 KB	18 Jul 2011	Paolo Bacchilega
exif-data-after-no-crash.txt	12.7 KB	18 Jul 2011	Paolo Bacchilega
exif-data-before-after-diff.txt	3.47 KB	19 Jul 2011	Paolo Bacchilega
exiv2-bug769.jpg	6.01 KB	19 Jul 2011	Andreas Huggel
exif-data-after.txt.cmd	11.4 KB	19 Jul 2011	Andreas Huggel