

Exiv2 - Bug #1195

Misclassified Canon Lens TagInfo

22 Jun 2016 07:12 - Robin Mills

Status:	Assigned	Start date:	22 Jun 2016
Priority:	Normal	Due date:	
Assignee:	Robin Mills	% Done:	10%
Category:	makernote	Estimated time:	20.00 hours
Target version:	0.28		

Description

I've been using your library in the context of implementing some improvements to the hdrmerge tool, an application written by Wenzel Jakob, available on github.

I am working on a way to do automatic TCA/Vignetting correction using the lensfun library.

This in turn relies on having access to the lens name, as provided by exiftool in its LensID tag (this I'm recovering using the write() method on the iterator returned by Exiv2::lensName(), per a suggestion from Andreas). The other thing it relies on is information about the focus distance, for Canon cameras this is available as a pair of integers (uint16) in the File Info block of the maker notes, number 0x14 and 0x15. I noticed these entries are misclassified in canonmn.cpp:1347 (the TagInfo is set up as signedShort instead of unsignedShort), so I went in and changed the TagInfo setup, but nothing seems to have changed in the returned result, somehow the Metadatum still has signedShort type. The I noticed that the canonFiCfg struct in tiffimage.cpp defaults to ttSignedShort entries, while seems to ignore the type provided in the TagInfo struct. Indeed changing the default decoded type to ttUnsignedShort changes the whole struct types around. So I'm guessing if you could help me find where the type overriding should engage, I could fix it (and if you like send you a patch of some kind).

History

#1 - 22 Jun 2016 07:35 - Robin Mills

- % Done changed from 0 to 20

- Estimated time set to 5.00 h

In general, exiv2 isn't a metadata policeman. He reports what he finds in an image, so I'm not surprised that changing TagInfo has no effect.

You may find the option -pR useful for exploring the data stored in an image. -pR = Recursively print image data structures embedded in the file. Perhaps you could provide a test file so that we are both "on the same page" in this discussion.

#2 - 22 Jun 2016 08:00 - Robin Mills

I've received this email from Lucas:

Are you saying that the type in the TagInfo is meant to be overridden by the default setup in canonFiCfg? Or are you saying that the type comes from the IFD data in the file?

In either case we have a bug: in the first case, the setup of the code is inappropriate, because Canon's File Info struct has a mixture of datatypes in it, and these are not being propagated correctly when the Exifmetadatum object is constructed (because of this overriding behaviour that seems defective). In the case in which the file should command the type, we also have a defect, because other evidence (exiftool, for example) suggests the type should be uint16 whereas still the struct's opinion dominates. So it seems to me, a way or another there is some further understanding to be gained.

And a second email

Further to this, I suspect I'm seeing further evidence of this being a problem:

If you look at tiffcomposite.cpp:1837, you can see how the fix for issue 1337 was put a patch in toTyped() to make it "understand" how pentax and nikon's tags such and such have a different signedness compared to their neighbours.

What I keep not understanding is why isn't it simpler to use the values in the tagInfo structures directly. It seems that this double-definition can only cause problems...

FWIW, at present the fix for this "bug" I found is simply to add another similar special case:

Canonmn.cpp:1347 and 1348: change signedShort to unsignedShort

Canonmn.cpp:1824, same change

Tiffcomposite.cpp:1851: add check much like the Nikon/Pentax trap already present:

```
If ti == signedShort
    If (tag == 0x14 or tag == 0x15) && group == canonFiId
```

```
Ti = unsignedShort
```

I suppose the actual issue is understand whether or not this is desirable behaviour or not.

My response

Please provide a test file. I didn't write Exiv2, so I have to read/debug/think/guess what's going on. If we are both dealing with the same data, that's one less source of confusion.

I see you are in NZ and I am in England. So no overlap in the working day. I'm available on Skype 'clanmills' until about 11pm in England (10am in NZ).

#3 - 24 Jun 2016 10:26 - Robin Mills

I spoke to Luca on Skype. Here's the transcript from the post-mortem (Time is BST = UTC+1.00)

23-Jun-2016, 22:57:55 **Robin Mills**: Call 35 minutes 54 seconds

23-Jun-2016, 23:01:13 **Robin Mills**: 35 minutes usefully spent. Right. You patch your v0.25 code and you're OK for now. Later in the year (September) we'll work on tiffvisitor and see if we can get him to enforce the data type. And you may be able to compare the exiftool tables with Exiv2/mumbleMN.cpp tables - although without us enforcing the data coercion, that's possibly of limited help. Anyway, let's regroup about this in (your) spring/ (my) fall.

23-Jun-2016, 23:02:58 **Luca Fascione**: Yes that's right. Thanks a lot it was great meeting you!

23-Jun-2016, 23:04:10 **Robin Mills**: The test-suite stuff is documented in several places such as: <http://www.exiv2.org/download.html> where Andreas says to run

```
$ svn export svn://dev.exiv2.org/svn/tags/0.25/test # to get the test files (34mb)
$ make tests # to run the tests themselves.
```

I've written a longer note about the test suite here: http://dev.exiv2.org/projects/exiv2/wiki/How_do_I_run_the_test_suite_for_Exiv2

23-Jun-2016, 23:06:29 **Luca Fascione**: Great thanks!

23-Jun-2016, 23:07:28 **Robin Mills**: You may email me directly and I will reply. I've no idea why you've had trouble with Redmine. Perhaps you could register with your gmail account or something. For sure, it's better to talk on the forum because I'm not the only person to read it. And replies are stored forever and can be searched. In this case, I had totally forgotten about this from 3 years ago, and yet you were able to see my previous words/work on the subject. So let's try to use the Forum. Better for everybody.

23-Jun-2016, 23:08:29 **Luca Fascione**: Yes. I'll see to make another account

23-Jun-2016, 23:08:46 **Robin Mills**: Good. Speak Later.

23-Jun-2016, 23:09:08 **Luca Fascione**: Best of luck there (y)

#4 - 24 Jun 2016 10:28 - Robin Mills

- % Done changed from 20 to 10

- Estimated time changed from 5.00 h to 20.00 h

#5 - 25 Jun 2016 05:28 - Andreas Huggel

How does Exiv2 determine Exif tag types?

1) When writing: For normal Exif tags, the type used in the Exifdatum will determine the type written to the file. For makernote tags, which do not store tag type information in their binary representation, the user has to make sure the Exifdatum type is compatible with the binary structure of the makernote tag. (As complicated as that may sound, it is not an issue.)

2) When reading: For normal Exif tags, the type from the Exif binary structure is used. For some makernote tags, the binary structure does not contain type information, so Exiv2 has to know it. Such makernote constructs are configured in the TIFF parser, in tiffimage.cpp. For example, the canonFiCf structure contains configurations for the entire array and in canonFiDef you can configure individual tags, if they are not all the same. It is there where you'd configure, among other things, if the 16 bit should be read as a signed or unsigned integer.

As camera vendors rarely provide makernote specifications, all these configurations have been reverse engineered. The best and most up-to-date understanding of makernote tags is usually found in the exiftool source code.

(I do not immediately understand why the toType() function had to be changed in tiffcomposite.cpp in [r2962](#) as pointed out above, maybe that should be revisited with these explanations in mind.)

3) The role of TagInfo: The type in the TagInfo structure is **only** used as the default type to create an Exifdatum when the user doesn't explicitly specify a type, e.g., in a statement like `exifData["Exif.Image.YResolution"] = "-2/3";` (and for the reference documentation on the website.)

Yes, the type in TagInfo is redundant in the case of binary makernote constructs without type info, like canonFiCf. For consistency, it has to correspond to the configuration in tiffimage.cpp.

#6 - 26 Jun 2016 21:52 - Luca Fascione

Hi Andreas,

thanks for the explanation, I will review my proposal patch accordingly (and while I'm at it, also recode [r2962](#) in the same way).

As much as I understand it won't be the most elegant thing in the world, what do you think about the idea of adding a piece of code somewhere that verifies the two definitions agree? Potentially conditioned to some flag, be that runtime or a pre-processor define of some kind. You know, just so that on the one side one could run a test and check it's all consistent, and on the other side, it'd be a very useful breadcrumb for somebody else trying to find their way around the code, to see a couple lines of code saying "this is what the tiff parser thinks, this is what the TagInfo layer thinks, they ought to agree".

Thanks again

Luca

#7 - 30 Jun 2016 07:20 - Andreas Huggel

An automatic way to verify that the types in the tag definitions of the TIFF parser and corresponding TagInfo types match would be a welcome check; it's not straightforward to check manually. Personally, I'd prefer it to be completely outside of the library code (e.g., a test script or program), as this is not something we need to ship to every exiv2 library user. And I suspect it will take more than a couple of lines of code. :)

To make the code more understandable, maybe I should put some of the explanations above in the code as comments and/or into the wiki.