

## Exiv2 - Bug #1203

### Exif.CanonCs.FocusContinuous reported as Short

11 Aug 2016 15:13 - Sridhar Boovaraghavan

<b>Status:</b>	Assigned	<b>Start date:</b>	11 Aug 2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Sridhar Boovaraghavan	<b>% Done:</b>	0%
<b>Category:</b>	makernote	<b>Estimated time:</b>	5.00 hours
<b>Target version:</b>	0.28		
<b>Description</b>			
<p>Please change it to signed short (follows your documentation, see <a href="http://www.exiv2.org/tags-canon.html">http://www.exiv2.org/tags-canon.html</a>) and it is being interpreted as an unsignedShort in the code. 65535 is the unsigned short version of -1 (signedShort).</p> <p>For reference, exiftool reports -1.</p> <p>All indications are that this is a bug in exiv2. It not being raised in 8 years possibly just means that no one was interested in that tag till now.</p> <p>As to what -1 actually means, that's a different question.</p> <p>I didn't have the camera on any special continuous focusing mode at that time. If we don't know what -1 means technically, at least we should create a new value, say Off for -1.</p> <p>For testing purposes, one could use exiv2's test files itself. Try exiv2\test\data\exiv2-test.out(Line #2697). See that it reports (65535) whereas it should be a -1.</p> <p>Regards, Sridhar</p>			
<b>Related issues:</b>			
Related to Exiv2 - Bug #1202: Exif.CanonCs.FocusContinuous = 8 = Manual		<b>Closed</b>	<b>11 Aug 2016</b>
Related to Exiv2 - Bug #1206: In specific Canon makernote tags, certain value...		<b>Assigned</b>	<b>13 Aug 2016</b>
Related to Exiv2 - Feature #894: Canon Highlight Tone Priority		<b>Assigned</b>	<b>17 Mar 2013</b>
Related to Exiv2 - Feature #1219: Adding Camera Temperature		<b>Assigned</b>	<b>31 Aug 2016</b>

### Associated revisions

#### Revision 4438 - 26 Aug 2016 05:36 - Sridhar Boovaraghavan

This is mainly a fix for #1206, but also interprets missing Canon Exif Tags in exiv2 with the help of Phil Harvey's exiftool (see <http://www.sno.phy.queensu.ca/~phil/exiftool/TagNames/Canon.html>).

Even with these changes (toward #1204 and #1205), exiv2 lags behind exiftool in some areas of interpretation of Canon tags. Ideally, a catch-up effort to bring the code in source: canonmn.cpp in line with lib/Image/ExifTool/Canon.pm. v10.25 of exiftool was used as reference for this change.

#1206 seeks to address the fact that when Canon does not have data for certain tags, they use specific default values in those fields. These default values need to be ignored and not displayed. This change brings this feature to exiv2, something that already exiftool does.

With regards to implementation, the struct TagInfo in source: tags.hpp is extended with four new fields.

The first field is a bool that if set to true (default false), denotes that this field has ignorable default values.

The second field is the default value that needs to be ignored. This can be of four types (String, Long, Float, Rational). These four types were chosen as they had conversion functions in the Value class.

The third field is the comparison type (default equal\_to). There are

six comparison types possible (equal\_to, not\_equal\_to, less, less\_equal, greater, greater\_equal). This is the comparison applied to the value stored in the tag's field and the default value specified above. For e.g. if the value in the tag Exif.CanonCs.RecordMode is -1, then it needs to be ignored.

The fourth field is the data type (default Long). This could have been guessed from the type of the second field, but that would necessitate making this structure into a template calling for changes in multitude of files.

Usage: In source: canonmn.cpp, several exif tags now have ignorable default properties. I will list a few examples.

1. Exif.CanonCs.FocusMode: TagInfo(0x0007, "FocusMode", N\_("Focus Mode"), N\_("Focus mode setting"), canonCsId, makerTags, signedShort, 1, EXV\_PRINT\_TAG(canonCsFocusMode)),

There are no changes - i.e. this is an example of how the TagInfo structure was being populated.

2. Exif.CanonCs.RecordMode: TagInfo(0x0009, "RecordMode", N\_("Record Mode"), N\_("Record mode setting"), canonCsId, makerTags, signedShort, 1, EXV\_PRINT\_TAG(canonCsRecordMode), true, s\_1\_),

Take a look at the two new arguments. The first one (true) specifies that there is a default value that can be ignored. The second one s\_1\_ specifies the value (-1, in this case) to be ignored.

```
const UShortValue CanonMakerNote::s_1_(65535, unsignedShort); // Till bug is resolved
```

Note s\_1\_ is temporarily having the value 65535 till #1203 that causes signedShorts to be interpreted as unsignedShorts is resolved.

3. Exif.CanonSi.TargetAperture: TagInfo(0x0004, "TargetAperture", N\_("Target Aperture"), N\_("Target Aperture"), canonSild, makerTags, unsignedShort, 1, printSi0x0015, true, us0\_, TagInfo::less\_equal),

Note the third argument TagInfo::less\_equal. This combined with the second argument us0\_ (the number 0) signifies that any values in this tag that are less than or equal (<=) to 0 should be ignored.

4. TagInfo(0x0028, "ImageUniqueID", N\_("Image Unique ID"), N\_("Image Unique ID"), canonId, makerTags, asciiString, -1, printValue, true, s0x16\_, TagInfo::equal\_to, TagInfo::String),

The previous examples have all been of Long type. This shows a case where the default value is a string.

```
const AsciiValue CanonMakerNote::s0x16_("0000000000000000");
```

Once these tag values have been defined, the actual mechanics of ignoring these default values happens in Image::exifData().

Before the exifData is returned, we loop through the data, ask the data whether it needs to be ignored (which in turn checks its underlying tagInfo and compares it with the default value, if specified) and if so, deletes that element.

A compile-time switch called EXV\_DONT\_IGNORE\_UNDEFINED which when set to a non-zero value will cause the behavior to revert back to the original where all values are reported irregardless of the fact that they need to be ignored.

#### **Revision 4446 - 30 Aug 2016 06:58 - Robin Mills**

#1203 Fixing test suite to respect changes introduced by r4438

#### **Revision 4447 - 30 Aug 2016 07:32 - Robin Mills**

#1203 test/webp-test.sh - silence warning about time format.

## **History**

---

### **#1 - 11 Aug 2016 15:42 - Robin Mills**

- Category set to makernote
- Status changed from New to Assigned
- Assignee set to Robin Mills

- Target version set to 0.28

- Estimated time set to 5.00 h

Sridhar

I'm happy to accept this bug and I've put it into v0.27 for resolution. I think there's something wrong in the makernote decoder/encoder. I'll look at this for the next release. I've updated the title of [#1102](#) to reflect the purpose of [r4365](#)

There's an additional artefact which I documented in [#1102](#). You can set this Key to a SShort from the command-line, however the value is reported as Short.

```
$ cp gnu/exiv2/trunk/test/tmp/20030925_201850.jpg cs.jpg
$ exiv2 -M"set Exif.CanonCs.FocusContinuous SShort -1" cs.jpg
$ exiv2 -pa -K Exif.CanonCs.FocusContinuous cs.jpg
Exif.CanonCs.FocusContinuous      Short      1      (65535)
```

The following warning is correct. -1 is an invalid Short (it's a SShort)

```
$ exiv2 -M"set Exif.CanonCs.FocusContinuous Short -1" cs.jpg
Warning: Exif.CanonCs.FocusContinuous: Failed to read Short value "-1"
```

#### **#2 - 11 Aug 2016 15:47 - Robin Mills**

- Subject changed from *Exif.CanonCs.FocusContinuous comes out as unsigned short to Exif.CanonCs.FocusContinuous reported as Short*

#### **#3 - 12 Aug 2016 05:45 - Sridhar Boovaraghavan**

There are other tags such as Exif.CanonCs.ImageStabilization (returns 65535), Exif.CanonCs.SpotMeteringMode (returns 65535), Exif.CanonCs.Sharpness (returns 32767) that are also incorrect.

They are also signed short in the exiv2 documentation and return shorts.

In summary, there seems to be a bigger bug, probably the same one that is affecting all these variables that incorrectly interprets signedShort variables as short.

I would appreciate this getting fixed soon after 0.26 if we are trying to get that out.

#### **#4 - 12 Aug 2016 07:00 - Sridhar Boovaraghavan**

Hi Robin,

Sorry for the multiple updates.

I feel like I have more information now.

When I ask exiftool to report on such images (where I get 65535), those tags are not reported unless you pass along a "-v" switch.

This along with those values not being defined (i.e. there is no interpretation for -1), makes me feel that these are tags that are not "written" and thus contain default values.

This reinforces my earlier suggestion of creating a -1, but interpreting it as a new key that we introduce - say Unknown.

I can confirm (for ImageStabilization at least), the lens in question does not have it (and neither does the camera). So when exiv2 returns -1 for that image for this field, it should be interpreted as Unknown. If you can think of a better name for this, please do suggest.

I had suggested "Off" in my original bug report above, but now feel that something like "Unknown" is better because it transcends categories like Exif.CanonCs.SpotMeteringMode and Exif.CanonCs.Sharpness.

Thanks,  
Sridhar

#### **#5 - 12 Aug 2016 08:56 - Robin Mills**

Sridhar. I'll put in a generic fix for this. I spent another 2 hours yesterday evening without figuring it out. It will have to wait for 0.27 because I'm fully already fully loaded for v0.26 and I took on another substantial task for v0.26 this week: [#1199](#).

I do not support your suggestion about interpreting -1 as "Unknown" or "Off". I don't want to introduce guesses. Do we reserve "Unknown" for -1, or for every value we don't know? We already have a way in which we report it and I want to leave it alone without having solid information. For example, I'm happy to change 8 to Manual based on Phil's website.

I'm always looking for engineers to help with Exiv2. Are you available and have good C++ skills?

#### **#6 - 12 Aug 2016 11:18 - Sridhar Boovaraghavan**

Hi Robin,

Yes, a generic fix is required in this case. Returning -1 and realizing that it is a signed short is better than the way it currently is.

Waiting for 0.27 is not a problem.

However, we need to understand why exiftool segregates between reporting -1 when verbose is specified and not reporting the value at all in the usual case. I feel that this is the crux of the issue. My guess is that FFFF is some code to designate all "Unknown" values - i.e. like when the makernote area is memset with FF and the other values are over written as they are discovered. I have raised this question in the exiftool forum (<http://u88.n24.queensu.ca/exiftool/forum/index.php/topic,7446.0.html>), let's see if we get any answers. We can look at the exiftool code to understand this as well, but my perl interpretation skills might not get us the answer quickly.

More interesting is the Sharpness case. The value returned is 32767 - a valid signed short, but reported as 3 by exiftool - a value that I recognize from the Canon menus as correct. Now, the verbose exiftool output shows 32767 - maybe a mistake there in that the data is not being interpreted but just reported blindly. Maybe that's how verbose is supposed to behave, but I don't like it. I would have preferred that the output was consistent in both cases and the "final" values reported would be the interpreted values and the raw output can be shown as additional data.

I am good in C++ and with some guidance on where to start looking can fix these kind of issues, but my bandwidth is uncertain. I am working on a personal project dealing with keeping my website in sync with my Picasa database and thus have gotten into interpreting tags etc. If you would like, you could make me a developer, but I can't promise reserved bandwidth to exiv2.

Thanks,  
Sridhar

#### **#7 - 12 Aug 2016 12:01 - Sridhar Boovaraghavan**

I guess my perl interpretation skills were enough (it was easy).

Please take a look at `Image-ExifTool-10.25/lib/Image/ExifTool/Canon.pm` in the exiftool source (I am using v10.25, latest as of today). Line #2016 (Sharpness) shows that 7FFF means undefined. Similarly line 2208 (SpotMeteringMode) clearly shows -1 is undefined, same for ImageStabilization (#2183) and FocusContinuous (#2163).

It is also apparent that FFFF (-1) means undefined for a lot of the other tags and 7FFF is also a special value for some of them.

Now, exiftool differentiates by reporting the RawValue (its terminology) in the verbose mode and not reporting an undefined value in normal mode. I feel that exiv2 should a "better" mix of the two - i.e. retrieve -1 correctly, but interpret it as undefined. This could mean an additional function in Exiv2::Metadatum that could let us know that the retrieved data is undefined and should not be used.

Regards,  
Sridhar

#### **#8 - 12 Aug 2016 12:26 - Robin Mills**

I'm happy to mentor you in the ways of the force. Adding stuff like 8/Manual is almost trivial ([#1202](#)). And we should add something to the test suite for [#1202](#). I'll probably do that over the weekend and then you'll have a model of what to do with stuff like that. I can't grant you write access to the Exiv2 repository, Andreas (the project founder) reserves that (and other) privileges. When new contributors volunteer, we normally ask them to provide patches for changes as we get to know each other.

[http://dev.exiv2.org/projects/exiv2/wiki/Contributing\\_to\\_Exiv2](http://dev.exiv2.org/projects/exiv2/wiki/Contributing_to_Exiv2)  
[http://dev.exiv2.org/projects/exiv2/wiki/Guidelines\\_for\\_submitting\\_patches](http://dev.exiv2.org/projects/exiv2/wiki/Guidelines_for_submitting_patches)

[#1203](#) is difficult. I spent a couple of hours on that last night. Not a job for a newbie.

I've added Phil (of exiftool fame) to the watchers of this bug. Phil: Can you give any insight concerning -1/unsignedShort metadata values?

#### **#9 - 12 Aug 2016 13:43 - Sridhar Boovaraghavan**

Sounds fine regarding contributing patches etc.

I have been corresponding with Phil separately on the same question. His answers are quoted below.

Summary: -1 is undefined and is not shown in exiftool output. 7fff is also undefined and is not shown. The sharpness value of 3 that I see in exiftool is actually Exif.CanonPr.Sharpness - or at least something like this. I say "something like this" because I do not see a CanonPr category in the exiv2 canon tags documentation. Though this is in the code.

I will use that going forward.

However, fixing Issue [#1203](#) the right way would involve introducing terminology like I suggest above to tell users that these values that are being returned are really not defined and should not be used. Or, we go the exiftool way and do not return a value at all - i.e. set the returned iterator to `ed.end()`. It might be easier to accomplish the latter.

Regards,  
Sridhar

Hi Sridhar,

Sorry, I missed this in the email, but I did respond about this in the forum post. Also, here is my response to the exiv2 bug email (but I'm not sure if an email response is valid for this bug list because it hasn't shown up on the web page yet):

On Aug 12, 2016, at 8:38 AM, Phil Harvey <phil@...> wrote:

Nothing further than what I mentioned here:

<http://u88.n24.queensu.ca/exiftool/forum/index.php/topic,7446.msg37852.html#new>

- Phil

> On Aug 12, 2016, at 8:08 AM, Sridhar Boovaraghavan <sridhar\_ml@...> wrote:

>

> Thanks, Phil.

>

> Could you please comment on the Sharpness case?

>

> After sending the original e-mail, I went to see the underlying perl source and see that Sharpness values of 7fff is undefined. How is that interpreted as 3 in the non-verbose output? Or is it another sharpness value that is being interpreted?

>

> Regards,

> Sridhar

>

>

> From: Phil Harvey <phil@...>

> To: Sridhar Boovaraghavan <sridhar\_ml@...>

> Sent: Friday, 12 August 2016 5:19 PM

> Subject: Re: ImageStabilization/Sharpness/SpotMetering - verbose output vs non-verbose

>

> Hi Sridhar,

>

> Some values in the metadata indicate "not applicable" or similar. Typically these are -1. The -v output shows these raw values, but ExifTool may sometimes choose to not generate a tag when this happens. For a bit more insight, see the Value Conversions section of this page:

>

> <http://owl.phy.queensu.ca/~phil/exiftool/under.html#conversions>

>

> - Phil

>

>> On Aug 12, 2016, at 3:26 AM, Sridhar Boovaraghavan <sridhar\_ml@...> wrote:

>>

>> Hi Phil,

>>

>> I would like to understand the difference between values returned with the verbose option and those without

>>

>> I am attaching the output of exiftool -a of two images along with the second file containing the output of exiftool -a -v3 on those same images.

>>

>> Specifically, I see ImageStabilization reported as -1 in the verbose output and not reported at all in the "normal" output.

>>

>> In both these images, ImageStabilization was not available and thus, I understand the "normal" output to be "correct".

>>

>> How am I to interpret the verbose output?

>>

>> Is FFFF some kind of default value stored in fields that are not used and thus those values should be "discarded" or ignored?

>>

>> For Sharpness, it is a more interesting question. The normal output returns a sharpness of 3 - which is correct. The verbose output returns 7FFF (or 32767). How is this translated to 3?

>>

>> Thanks,

>> Sridhar

#### #10 - 13 Aug 2016 05:20 - Sridhar Boovaraghavan

Here is a list of explicit tag values that exiftool ignores. The resolution of this issue should address this as well.

Note that this doesn't encompass other cases where values that are <= 0 are being ignored. Ideally, [source:canonmn.cpp](#) should do the latter as well. I think it does do this in many cases, but a check with Canon.pm of exiftool might show up cases that it doesn't.

```

Exif.CanonCs.RecordMode      -1
Exif.CanonCs.Contrast       0x7fff
Exif.CanonCs.Saturation     0x7fff
Exif.CanonCs.Sharpness      0x7fff
Exif.CanonCs.CameraISO     0x7fff
Exif.CanonCs.AFPoint        0
Exif.CanonCs.FlashActivity  -1
Exif.CanonCs.FocusContinuous -1
Exif.CanonCs.AESetting      -1
Exif.CanonCs.ImageStabilization -1
Exif.CanonCs.SpotMeteringMode -1
Exif.CanonCs.PhotoEffect    -1
Exif.CanonCs.ColorTone      0x7fff
Exif.CanonCs.SRAWQuality    -1
Exif.CanonSi.FlashGuideNumber -1
Exif.CanonSi.AFPointsInFocus 0
Exif.CanonMi.FrameRate      65535
Exif.CanonMi.FrameCount     65535
Exif.CanonFi.FilterEffect   -1
Exif.CanonFi.ToningEffect   -1

```

#### #11 - 13 Aug 2016 08:53 - Robin Mills

Are you asking for those tags to be ignored by Exiv2, or to be not reported when they have those specific values?

We seem to be discussing several different things simultaneously:

- 1) The signed/unsigned conflict
- 2) How to "translate" undefined values such as -1
- 3) The list of tags above

Can you open different issues for those topics. Redmine has a feature for "Related Issues" which can be used to group these matters together. You may wish to open an "Umbrella" which is a virtual issue to collect numerous related issues. The title could be "Umbrella: Canon MakerNote Undefined Values"

#### #12 - 13 Aug 2016 10:01 - Sridhar Boovaraghavan

As requested, I have created Issue [#1206](#) to cover the case of ignoring specific values for specific Canon makernote tags.

#2 and #3 should be addressed by Issue [#1206](#).

It will be good if Issue [#1203](#) stuck to resolving why these tags return unsigned short and fixing that.

#### #13 - 30 Aug 2016 07:00 - Robin Mills

- Assignee changed from Robin Mills to Sridhar Boovaraghavan

- Target version changed from 0.28 to 0.26

[r4438](#) Thank you, Sridhar for taking care of this.

[r4446](#) Fixes to test suite exceptions introduced by [r4438](#). I believe these are of two types:

- 1) Intended changes in product behaviour
- 2) Changes being introduced by the *diff* utility in creating the reports.

There are differences being reported in the tags Exif.Image.ExifTag and Exif.Photo.MakerNote. I've done a comparison of 525 tags in the 350 jpgs in trunk/test/data using both --revision 4436 and HEAD (--revision 4444). There are no differences. These reports are being generated by diff being confused by the interspersed of those tags with the CanonXy makernote decoded/generated tags which are no longer reported by [#1203](#). In other words, the mentions of ExifTag and MakerNote are artefacts of the test script.

```

811 rmills@rmillsmbp:~/gnu/exiv2/t4436 $ t=$(basename $PWD); exiv2 -vVg svn; \
find ../trunk/test -name "*.jpg" | xargs exiv2 -pa --grep MakerNote --grep ExifTag 2>/dev/null >/tmp/$t.txt ;
echo $t $(wc /tmp/$t.txt)
exiv2 0.25 001900 (64 bit build)
svn=4436
t4436 525 3444 57619 /tmp/t4436.txt
812 rmills@rmillsmbp:~/gnu/exiv2/t4436 $ cd ../trunk
813 rmills@rmillsmbp:~/gnu/exiv2/trunk $ makes
make
      7      57      583
sudo make install
     108     601     7939
814 rmills@rmillsmbp:~/gnu/exiv2/trunk $ t=$(basename $PWD); exiv2 -vVg svn; \
find ../trunk/test -name "*.jpg" | xargs exiv2 -pa --grep MakerNote --grep ExifTag 2>/dev/null >/tmp/$t.txt ;
echo $t $(wc /tmp/$t.txt)

```

```
exiv2 0.25 001900 (64 bit build)
svn=4444
trunk 525 3444 57619 /tmp/trunk.txt
815 rmills@rmillsmbp:~/gnu/exiv2/trunk $ diff /tmp/t4436.txt /tmp/trunk.txt
816 rmills@rmillsmbp:~/gnu/exiv2/trunk $
```

The following anomalies are still to be fixed:

- 1 write-test.sh is reporting concerning Exif.CanonSi.MeasuredEV2
- 2 imagetest.sh is reporting an issue concerning smiley2.jpg.c3tst
- 3 webp-test.sh is reporting: < Warning: Unsupported time format

I will take responsibility for resolving and eliminating those issues with the test suite. When I fix them, I'll close and assign this issue to you to be certain that the v0.26 release notes credit you with the changes introduced by [#1203](#) (and related issues).

**#14 - 30 Aug 2016 08:37 - Sridhar Boovaraghavan**

Sorry, Robin. I think you are talking about [#1204](#) and [#1205](#).

Thanks for offering to update the canonical test results to match the newly-set defaults that ignore undefined values.

However, [#1203](#) is still a problem. We have requested Andreas to address this.

Thanks,  
Sridhar

**#15 - 30 Aug 2016 10:42 - Robin Mills**

The investigation of the test harness anomalies introduced by [r4438](#) is being deal with in [#1218](#).

**#16 - 30 Aug 2016 16:11 - Robin Mills**

- Target version changed from 0.26 to 0.28

Development of this feature has been deferred for v0.27 and to be developed in branches/develop. [r4449](#)