

Exiv2 - Bug #1206

In specific Canon makernote tags, certain values are meant to be ignored

13 Aug 2016 09:55 - Sridhar Boovaraghavan

Status:	Assigned	Start date:	13 Aug 2016
Priority:	Normal	Due date:	
Assignee:	Sridhar Boovaraghavan	% Done:	0%
Category:	metadata	Estimated time:	0.00 hour
Target version:	0.28		
Description			
<p>We already saw in Issue #1203 that certain canon makernote tags were returning values of incorrect type.</p> <p>Further investigation (basically, a comparison with exiftool, see notes attached to Issue #1203) make it clear that some of the returned values are defaults that are meant to be ignored.</p> <p>A selection of these are listed below. The table contains the tag name and the value which if it has, should be ignored.</p> <p>There are other tag values that need to be ignored as well, but those are in the form of values needing to be >0 etc.</p> <p>A good start would be to implement a filter type function that defines for each tag what an ignorable value means. If this function returns true, then we return ed.end() when the user calls findMetadatum.</p> <p>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</p>			
Related issues:			
Related to Exiv2 - Bug #1203: Exif.CanonCs.FocusContinuous reported as Short		Assigned	11 Aug 2016
Related to Exiv2 - Feature #1219: Adding Camera Temperature		Assigned	31 Aug 2016

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"), canonSiId, 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.

History

#1 - 13 Aug 2016 10:02 - Sridhar Boovaraghavan

- Priority changed from Normal to High

#2 - 13 Aug 2016 10:21 - Robin Mills

- Category set to metadata
- Status changed from New to Assigned
- Assignee set to Sridhar Boovaraghavan
- Priority changed from High to Normal
- Target version set to 0.26

Are these keys to be always ignored, or only when they have those default values? Should we report those values if the user provides the `--verbose` option on the command-line?

#3 - 13 Aug 2016 11:37 - Sridhar Boovaraghavan

No, these keys are not always meant to be ignored, but only when they have those specific values.

Regarding the verbose option, I don't know. Does `exiv2` do anything different as in print the raw (uninterpreted) values in addition to the interpreted ones? If so, then we can consider printing those values when verbose is used.

#4 - 13 Aug 2016 12:45 - Robin Mills

I don't think verbose does much. And it's a feature of `exiv2(.exe)` and not the library. If you decide to remove those in `canonmn.cpp`, the `--verbose` flag will not be able to see those keys.

`--verbose` gives a little more output by reporting file paths and stuff.

```
954 rmills@rmillssmbp:~/gnu/exiv2/trunk $ exiv2 -pv --grep LensID/i ~/Stonehenge.jpg
0x000c NikonLd3      LensIDNumber      Byte      1 146
955 rmills@rmillssmbp:~/gnu/exiv2/trunk $ exiv2 -pa --grep LensID/i ~/Stonehenge.jpg
Exif.NikonLd3.LensIDNumber      Byte      1 Sigma 18-250mm F3.5-6.3 DC OS Macro HSM
956 rmills@rmillssmbp:~/gnu/exiv2/trunk $ exiv2 --verbose -pa --grep LensID/i ~/Stonehenge.jpg
File 1/1: /Users/rmills/Stonehenge.jpg
Exif.NikonLd3.LensIDNumber      Byte      1 Sigma 18-250mm F3.5-6.3 DC OS Macro HSM
957 rmills@rmillssmbp:~/gnu/exiv2/trunk $ exiv2 --verbose -pv --grep LensID/i ~/Stonehenge.jpg
File 1/1: /Users/rmills/Stonehenge.jpg
0x000c NikonLd3      LensIDNumber      Byte      1 146
958 rmills@rmillssmbp:~/gnu/exiv2/trunk $
```

`--verbose` also reports when the return value of `exiv2(.exe)` is not zero.

A very important use of `--verbose` is with `--version`. It prints a lot of information about the build **AND** the paths to dynamic libraries loaded by `exiv2(.exe)`.

#5 - 30 Aug 2016 16:13 - 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](#)

#6 - 31 Aug 2016 15:11 - Andreas Huggel

Based on the list above, all of these tags are in 'binary arrays', i.e., composite IFD tags which consist of a number of values that are decoded into different tags, e.g., a single IFD tag consisting of three short values 1, 2, 3 which are decoded into three different tags, tag1 with value short 1, tag2 with value short 2 and tag3 with value short 3.

My understanding is that Canon decided to use a certain value to identify tags which are not present in certain images (maybe because the camera used doesn't support the feature related to the particular tag). So, e.g., the second value of the above sample array could be set to -1 to indicate that tag2 is not present, as, because of the choice to use a binary array for the tags, that tag cannot simply be omitted from the binary structure.

`Exiv2` should ignore such tags, as if they were really not present in the image.

A straightforward implementation will be to enhance the TIFF parser to not read such tags at all:

New settings should be added to `Internal::ArrayCfg` to make it possible to define the tags to be ignored. The actual configurations for the known arrays are in `tiffimage.cpp`. `TiffReader::visitBinaryArray` can then use the new configuration to skip these tags.

The write code may require minor changes as well.