Exiv2 - Feature #1039

Wrong ApertureValue written

09 Mar 2015 20:40 - Torsten Flammiger

Status:	Closed	**Start date:**	09 Mar 2015
Priority:	Normal	**Due date:**	
Assignee:	Robin Mills	**% Done:**	100%
Category:	samples	**Estimated time:**	0.00 hour
Target version:	0.25		

Description

Hello

I expect that programs like exiftool or Irfanview show the correct values after I tagged them
with exiv2 library. For instance, I tag a jpg file with f/5.6 like so:

```
// works, shows F5.6
exifData["Exif.Image.FNumber"] = Exiv2::Rational(56, 10);

// but this does not work: shows F6.7 (rounded, the real value is 6.96)
// applies also to MaxApertureValue
exifData["Exif.Image.ApertureValue"] = Exiv2::Rational(56, 10);
exifData["Exif.Image.MaxApertureValue"] = Exiv2::Rational(56, 10);
```

Is there a special reason for this behavior? The value is multiplied by a half stop I think.
I'm on x64 Windows 8.1 and compiled both the library version 0.24 and my Shell extension with VS2012 as 64bit binary.

Torsten

Associated revisions

Revision 3621 - 11 Mar 2015 10:55 - Robin Mills

#1039. Thank You, Torsten for raising this matter. Thank You, Phil for your help with this.

Revision 3622 - 11 Mar 2015 16:03 - Robin Mills

#1039 Fixed typos in r3621

Revision 3623 - 11 Mar 2015 16:13 - Robin Mills

#1039 Fixed another typo in r3621

History

#1 - 09 Mar 2015 21:08 - Robin Mills

- Category set to metadata

- Status changed from New to Assigned

- Assignee set to Robin Mills

- Target version set to 0.25

Torsten

Three things come instantly to mind about this:

1. I had a discussion about Rational values with Mikayel last week. http://dev.exiv2.org/boards/3/topics/1898
2. You'll see that I offered Exiv2::floatToRationalCast(float); for setting a rational value
3. Are you sure this isn't a presentation matter and the string '6.7' is a mis-representation of the stored data.

To illustrate this, I've listed the FNumber in an image using the v (value) and t (translated) value:

```
506 rmills@rmillsmbp:~ $ exiv2 -pv -g FNumber ~/R.jpg
0x829d Photo          FNumber                   Rational   1 35/10
507 rmills@rmillsmbp:~ $ exiv2 -pt -g FNumber ~/R.jpg
Exif.Photo.FNumber                              Rational   1 F3.5
```

```
508 rmills@rmillsmbp:~ $
```

And to illustrate the point, I've tried to set the FNumber to 6.96

```
515 rmills@rmillsmbp:~ $ exiv2 -M"set Exif.Photo.FNumber 696/100" ~/R.jpg
516 rmills@rmillsmbp:~ $ exiv2 -pv -g FNumber ~/R.jpg
0x829d Photo        FNumber                   Rational    1   696/100
517 rmills@rmillsmbp:~ $ exiv2 -pt -g FNumber ~/R.jpg
Exif.Photo.FNumber                            Rational    1   F7
```

I looks to me as though it's being format as printf's "%f.1" (floating point with one decimal place).

If I try to set 694 (694/100), the "translated" value is 6.9

```
526 rmills@rmillsmbp:~ $ exiv2 -M"set Exif.Photo.FNumber Rational 694/100" ~/R.jpg
527 rmills@rmillsmbp:~ $ exiv2 -pv -g FNumber ~/R.jpg
0x829d Photo        FNumber                   Rational    1   694/100
528 rmills@rmillsmbp:~ $ exiv2 -pt -g FNumber ~/R.jpg
Exif.Photo.FNumber                            Rational    1   F6.9
529 rmills@rmillsmbp:~ $
```

#2 - 09 Mar 2015 21:30 - Torsten Flammiger

Hmm, I played a little bit with the exif2 executable v0.24 I just downloaded to avoid errors I might have done myself in handling the library. And all I can say is that the values for ApertureValue and MaxApertureValue I write into the file with exiv2.exe are not the same I read from the jpeg file. However, the value stays the same if I use FNumber.

exiv2 -M"set Exif.Image.FNumber 28/10" 20150309-DSC04368.jpg
exiv2 -pt -g FNumber 20150309-DSC04368.jpg

1. Exif.Image.FNumber SRational 1 F2.8

but:

exiv2 -M"set Exif.Image.MaxApertureValue 28/10" 20150309-DSC04368.jpg
exiv2 -pt -g MaxApertureValue 20150309-DSC04368.jpg
Exif.Image.MaxApertureValue SRational 1 F2.6

I can't imagine that this has something to do with rounding errors through printf's "%f.1"

#3 - 09 Mar 2015 21:43 - Robin Mills

You're right. Gosh, that can't be presentation. (BTW, we don't use printf).

```
535 rmills@rmillsmbp:~ $ exiv2 -M"set Exif.Image.MaxApertureValue 28/10" -M"set Exif.Image.FNumber 28/10" ~/R.
jpg
536 rmills@rmillsmbp:~ $ exiv2 -pt -g Image.MaxApertureValue -g Image.FNumber ~/R.jpg
Exif.Image.FNumber                            Rational    1   F2.8
Exif.Image.MaxApertureValue                   Rational    1   F2.6
537 rmills@rmillsmbp:~ $ exiv2 -pv -g Image.MaxApertureValue -g Image.FNumber ~/R.jpg
0x829d Image        FNumber                   Rational    1   28/10
0x9205 Image        MaxApertureValue          Rational    1   28/10
538 rmills@rmillsmbp:~ $
```

One of us will have to walk the code in Visual Studio's debugger and find out what's cooking here. *Can I ask you* to do that and save me a task?

I'm going to add Niels to the watcher's list for this issue. Niels is our makernote engineer and might know about this.   Niels is a busy school teacher, so don't expect him to jump on this.

#4 - 09 Mar 2015 22:00 - Torsten Flammiger

Unfortunately I am unable to build the exe by myself. There is a problem with zlib in my build
and the executable relies on zlib to support png files (if I understand that correctly).
I will try to fix this later, but not today. It's too late now in Germany ;)

#5 - 09 Mar 2015 22:20 - Robin Mills

Well it's getting late in England and I've been working for about 12 hours on Exiv2 and another open-source project. I'll debug this tomorrow.

I've started with an image with no-meta data.

```
543 rmills@rmillsmbp:~ $ exiv2 -pa ~/X.jpg
/Users/rmills/X.jpg: (No XMP data found in the file
48 rmills@rmillsmbp:~ $ exiv2 -M"set Exif.Image.MaxApertureValue 28/10" -M"set Exif.Image.FNumber 28/10" ~/X.j
pg
```

```
549 rmills@rmillsmbp:~ $ exiv2 -pv ~/X.jpg
0x829d Image        FNumber                   Rational    1  28/10
0x9205 Image        MaxApertureValue          Rational    1  28/10
550 rmills@rmillsmbp:~ $ exiv2 -pt ~/X.jpg
Exif.Image.FNumber                        Rational    1  F2.8
Exif.Image.MaxApertureValue               Rational    1  F2.6
```

I only need to step through the last command to find out why the values presented by the -pt option are different.

If I don't get back to you in a couple of days, "ping me" as I might forget. I had a new grandson born last week and lots of other stuff happening at the moment.

#6 - 10 Mar 2015 10:21 - Robin Mills

I've found the reason for MaxApertureValue being displayed as F2.6 for Rational 28/10. The value 2.8 is filtered by this function in tags.cpp

```
    float fnumber(float apertureValue)
    {
        return static_cast<float>(std::exp(std::log(2.0) * apertureValue / 2));
    }
```

which is called by the function to print that tag:

```
    std::ostream& print0x9202(std::ostream& os, const Value& value, const ExifData*)
    {
        std::ios::fmtflags f( os.flags() );
        if (   value.count() == 0
            || value.toRational().second == 0) {
            return os << "(" << value << ")";
        }
        std::ostringstream oss;
        oss.copyfmt(os);
        os << "F" << std::setprecision(2) << fnumber(value.toFloat());
        os.copyfmt(oss);
        os.flags(f);
        return os;
    }
```

Why do we do this? I don't know because I'm not a photographer. I've checked the Exif Specification and it doesn't say anything special about MaxApertureValue. http://www.kodak.com/global/plugins/acrobat/en/service/digCam/exifStandard2.pdf

I've looked one of my photographs from my Nikon 5300:

```
524 rmills@rmillsmbp:~/gnu/exiv2/trunk $ exiv2 -pv -g MaxApertureValue ~/Pictures/2015/WinterTrip/Austin/DSC_5
850.jpg
0x9205 Photo        MaxApertureValue          Rational    1  43/10
525 rmills@rmillsmbp:~/gnu/exiv2/trunk $ exiv2 -pt -g MaxApertureValue ~/Pictures/2015/WinterTrip/Austin/DSC_5
850.jpg
Exif.Photo.MaxApertureValue               Rational    1  F4.4
526 rmills@rmillsmbp:~/gnu/exiv2/trunk $
```

You can see the value F4.4 when the data seems to indicate F4.3.

The function float fnumber(float apertureValue) is called in 3 files: cannon.cpp, crwimage.cpp and tags.cpp. So there must be a relationship between the stored data and the presentation value.

I'm going to add Phil and Andreas to the watchers of this issue as they may be able to explain this. And I have two suggestions for you:

1. Replicate this behaviour with ExifTool and ask on their Forum.
2. Start Googling and discover more about Aperture/FNumbers

I'm not going to mark this "Resolved" at the moment, however I don't believe I can do anything else about this.

Here's a graph of the non-linear function:
 fnumber.png

#7 - 10 Mar 2015 10:39 - Robin Mills

Final thought about this. I've run my image on ExifTool and exiv2. The results are the same:

```
533 rmills@rmillsmbp:~/clanmills $ ExifTool ~/Pictures/2015/WinterTrip/Austin/DSC_5850.jpg  | grep -i aperture
Max Aperture Value          : 4.4
AF Aperture                 : 4.6
Max Aperture At Min Focal    : 3.6
Max Aperture At Max Focal    : 6.3
```

```
Effective Max Aperture          : 4.5
Aperture                        : 4.5
534 rmills@rmillsmbp:~/clanmills $ exiv2 -pt ~/Pictures/2015/WinterTrip/Austin/DSC_5850.jpg  | grep -i apertur
e
Exif.Photo.MaxApertureValue             Rational   1  F4.4
Exif.NikonLd3.AFAperture                Byte       1  F4.6
Exif.NikonLd3.MaxApertureAtMinFocal     Byte       1  F3.6
Exif.NikonLd3.MaxApertureAtMaxFocal     Byte       1  F6.3
Exif.NikonLd3.EffectiveMaxAperture      Byte       1  F4.5
535 rmills@rmillsmbp:~/clanmills $ $ exiv2 -pv ~/Pictures/2015/WinterTrip/Austin/DSC_5850.jpg  | grep -i apert
ure
0x9205 Photo        MaxApertureValue         Rational   1  43/10
0x0005 NikonLd3     AFAperture               Byte       1  53
0x0010 NikonLd3     MaxApertureAtMinFocal    Byte       1  44
0x0011 NikonLd3     MaxApertureAtMaxFocal    Byte       1  64
0x0013 NikonLd3     EffectiveMaxAperture     Byte       1  52
536 rmills@rmillsmbp:~/clanmills $
```

So there is method in the madness!

#8 - 10 Mar 2015 11:03 - Phil Harvey

The "ApertureValue" tags are stored as APEX values but converted to F-stops for display. There are two reasons for doing this: 1) F-stops are much more familiar than APEX values, and 2) This facilitates copying between aperture tags stored in different representations.

#9 - 10 Mar 2015 11:06 - Robin Mills

- Status changed from Assigned to Resolved

Thank You, Phil. I knew you'd know! Thank you for taking the time to explain this.

Torsten:
I'm going to mark this "Resolved".

#10 - 10 Mar 2015 17:39 - Torsten Flammiger

Thanks very much for taking the time and look into it.
Although I was not very satisfied with your answers ;) you pointed me in the right direction.

So, instead of tossing the real FNumber into the EXIV2::Rational(x, y) to tag the
image with ApertureValue and MaxApertureValue I now do something like this:

```
...
double d = std::floor( ((std::log(F-STOP-VALUE) / std::log(2)) * 2) + 0.5);
int numerator = static_cast<int>(d) * 10;
exifData["Exif.Image.ApertureValue"] = Exiv2::Rational(numerator, 10);
...
```

That way all works flawlessly (well, I ignore those little rounding errors...)
I tested this against exiv2 (yes, build now with the executable), exiftool and Irfanview.

Torsten

#11 - 10 Mar 2015 17:58 - Phil Harvey

I wouldn't be satisfied either if I had to do the APEX conversion manually. One might think that Exiv2 should do this automatically (to make reading and writing symmetrical).

#12 - 10 Mar 2015 19:05 - Robin Mills

I'm not sure how to turn your dissatisfaction into action to make you happy!

If I were to modify exiv2 to parse Ffloat -> Rational(numerator,10) using your function, would that help?

```
622 rmills@rmillsmbp:/Applications $ exiv2 -pa -g Photo.MaxApertureValue ~/R.jpg
Exif.Photo.MaxApertureValue             Rational   1  F3.5
623 rmills@rmillsmbp:/Applications $ exiv2 -M"set Exif.Photo.MaxApertureValue F4.3" ~/R.jpg
Warning: Exif.Photo.MaxApertureValue: Failed to read Rational value "F4.3"
624 rmills@rmillsmbp:/Applications $ exiv2 -pa -g Photo.MaxApertureValue ~/R.jpg
Exif.Photo.MaxApertureValue             Rational   1  F3.5
625 rmills@rmillsmbp:/Applications $
```

This is intended for keys such as MaxApertureValue, however it'll work on any key that accepts Rational.

Is that the fix that you would like to see implemented?

#13 - 10 Mar 2015 19:21 - Torsten Flammiger

Hmm, I am absolutely happy with my solution ;-) So there is no need to fix something I was probably the only one who has a problem with. I shared my *solution* so it migth be helpful for somebody else.

#14 - 10 Mar 2015 23:27 - Robin Mills

- Status changed from Resolved to Assigned

I'm pleased that you're happy with your solution. And thank you for sharing for others to learn from your knowledge.

I used your code to modify the Rational parser in types.cpp. The code *if (is.peek) ... } else* is new and parses the token Fnumber

```
    std::istream& operator>>(std::istream& is, URational& r)
    {
        uint32_t nominator;
        uint32_t denominator;
        // http://dev.exiv2.org/boards/3/topics/1912?r=1915
        if ( is.peek() == 'F' ) {
            char   F;
            double f;
            double d = 1000;
            is >> F >> f ;
            double   n = std::floor( 0.5 + d * 2.0 * std::log(f) / std::log(2.0) );
            nominator   = static_cast<int>(n) ;
            denominator = static_cast<int>(d);
        //  std::cout << "f = " << f << " n = " << n << " => " << nominator << "/" << denominator << std::endl
;
        } else {
            char c('\0');
            is >> nominator >> c >> denominator;
            if (c != '/') is.setstate(std::ios::failbit);
        }
        if ( is ) r = std::make_pair(nominator, denominator);
        return is;
    }
```

I've chosen to use the denominator of 1000 instead of your 10 as it seems to convince exiv2 to report the same Fx.y in the function number(float). I think we need to add an + 0.05 to that function to get input == output. I'll look at that tomorrow. Here's some output.

```
exiv2 "-Mset Exif.Photo.MaxApertureValue F22" y:\X.jpg && ^
exiv2 -pv -g Photo.MaxApertureValue          y:\X.jpg && ^
exiv2 -pa -g Photo.MaxApertureValue          y:\X.jpg
0x9205 Photo        MaxApertureValue          Rational    1  8919/1000
Exif.Photo.MaxApertureValue               Rational    1  F22

exiv2 "-Mset Exif.Photo.MaxApertureValue F4.3" y:\X.jpg && ^
exiv2 -pv -g Photo.MaxApertureValue          y:\X.jpg && ^
exiv2 -pa -g Photo.MaxApertureValue          y:\X.jpg
0x9205 Photo        MaxApertureValue          Rational    1  4209/1000
Exif.Photo.MaxApertureValue               Rational    1  F4.3

exiv2 "-Mset Exif.Photo.MaxApertureValue F2.6" y:\X.jpg && ^
exiv2 -pv -g Photo.MaxApertureValue          y:\X.jpg && ^
exiv2 -pa -g Photo.MaxApertureValue          y:\X.jpg
0x9205 Photo        MaxApertureValue          Rational    1  2757/1000
Exif.Photo.MaxApertureValue               Rational    1  F2.6
```

I'm going to set the status of this issue to "Assigned" as I intend additional work.

#15 - 10 Mar 2015 23:50 - Phil Harvey

Hi Robin,

Your solution sounds good, but the implementation in types.cpp level worries me because it sounds too general. I am not familiar with the Exiv2 code, but you should be sure that the APEX conversion applies only to the ApertureValue tags.

#16 - 11 Mar 2015 00:11 - Robin Mills

Thanks, Phil. You are right. It is general and I'm extending the syntax of Rational to be either n/m or Fnumber. I'll document this in the man page and explain about the APEX values. Anyone using the Fnumber syntax for anything other than aperture does so at their own peril. The current Exiv2 will report an error for Fnumber, so no existing scripts will be broken by this change.

I could of course add a flag in the code to say which keys accept the Fnumber syntax, however I don't believe we need this. If I clearly document

Fnumber, this should be sufficient.

#17 - 11 Mar 2015 11:05 - Phil Harvey

Sounds good to me.

#18 - 11 Mar 2015 11:18 - Robin Mills

Fix submitted. r3621 and r3622 Thank You to Torsten for reporting this issue. Thank You to Phil for his explanation and encouragement.

I think it's OK to provide the Fnumber syntax for general use because man page already says:

> It is possible to write tags with types and values different from those
> specified in the standards, duplicate Exif tags, undefined tags, or incomplete metadata.
> While **exiv2** is able to read all metadata that it can write, other programs may have difficulties
> with images that contain non standard-conforming metadata.

The submitted code avoids struggles with the denominator as follows:

```
std::istream& operator>>(std::istream& is, Rational& r)
{
    // http://dev.exiv2.org/boards/3/topics/1912?r=1915
    if ( std::tolower(is.peek()) == 'f' ) {
        char  F;
        float f;
        is >> F >> f ;
        f  = 2.0f * std::log(f) / std::log(2.0f) ;
        r  = Exiv2::floatToRationalCast(f);
    } else {
        int32_t nominator;
        int32_t denominator;
        char c('\0');
        is >> nominator >> c >> denominator;
        if (c != '/') is.setstate(std::ios::failbit);
        if (is) r = std::make_pair(nominator, denominator);
    }
    return is;
}
```

This works well over the range of Fnumbers available on my Nikon5300 with Sigma18-250 Lens.

```
755 rmills@rmillsmbp:~/gnu/exiv2/trunk $ for f in 2.6 5.6 7.1 13 22; do \
  exiv2 "-Mset Exif.Photo.MaxApertureValue F$f" ~/X.jpg  \
  exiv2 -pv -g Photo.MaxApertureValue ~/X.jpg \
  exiv2 -pa -g Photo.MaxApertureValue ~/X.jpg \
done
0x9205 Photo        MaxApertureValue           Rational   1  2757023/1000000
Exif.Photo.MaxApertureValue                    Rational   1  F2.6
0x9205 Photo        MaxApertureValue           Rational   1  2485427/500000
Exif.Photo.MaxApertureValue                    Rational   1  F5.6
0x9205 Photo        MaxApertureValue           Rational   1  2827819/500000
Exif.Photo.MaxApertureValue                    Rational   1  F7.1
0x9205 Photo        MaxApertureValue           Rational   1  92511/12500
Exif.Photo.MaxApertureValue                    Rational   1  F13
0x9205 Photo        MaxApertureValue           Rational   1  557429/62500
Exif.Photo.MaxApertureValue                    Rational   1  F22
756 rmills@rmillsmbp:~/gnu/exiv2/trunk $
```

I've updated the man page as follows:

The format for **Rational** (and *SRational*) is one of:

- integer
- integer-numerator/integer-denominator
- Number
- fnumber

Examples:

$ exiv2 "-Mset Exif.Photo.MaxApertureValue 557429/62500" X.jpg
$ exiv2 "-Mset Exif.Photo.MaxApertureValue F5.6" X.jpg

The format Fnumber is for the convenience of setting aperture values. The aperture value stored in Exif is an **APEX** value which can be evaluated by the expression::

- apex-value = log(Fnumber) * 2.0 / log(2.0)
- Fnumber = exp(apex-value * log(2.0) / 2)

The Fnumber syntax is valid for any **Rational**, even when the key is not an Aperture.

More information about **APEX** is available from: http://en.wikipedia.org/wiki/APEX_system

#19 - 11 Mar 2015 19:00 - Robin Mills

- Tracker changed from Support to Feature

- Category changed from metadata to samples

- Status changed from Assigned to Resolved

#20 - 08 May 2015 16:12 - Robin Mills

- % Done changed from 0 to 100

#21 - 21 Jun 2015 16:39 - Andreas Huggel

- Status changed from Resolved to Closed