

Exiv2 - Bug #1212

Sigma 18-35mm f/1.8 not recognized

25 Aug 2016 01:48 - Martin Ramshaw

Status:	Closed	Start date:	25 Aug 2016
Priority:	Normal	Due date:	
Assignee:	Thomas Beutlich	% Done:	100%
Category:	lens	Estimated time:	2.00 hours
Target version:	0.26		
Description			
Looks like Sigma has reused code 150 for this lens.			
[Filed as a Bug although it really seems like an enhancement to me.]			
It profiles as follows:			
owner@G30AB:/Photography/CR2s/2016-06-02\$ exiv2 -pt IMG_0634.CR2 grep -ai lens			
Exif.CanonCs.LensType Short 1 150			
Exif.CanonCs.Lens Short 3 18.0 - 35.0 mm			
Exif.Canon.LensModel Ascii 74 18-35mm			
Exif.Photo.LensSpecification Rational 4 18/1 35/1 0/1 0/1			
Exif.Photo.LensModel Ascii 8 18-35mm			
Exif.Photo.LensSerialNumber Ascii 11 0000000000			
owner@G30AB:/Photography/CR2s/2016-06-02\$			

History

#1 - 25 Aug 2016 06:05 - Thomas Beutlich

- Status changed from New to Assigned
- Assignee set to Thomas Beutlich
- Target version set to 0.26

It says "Canon EF 14mm f/2.8L or Sigma Lens" on <http://www.sno.phy.queensu.ca/~phil/exiftool/TagNames/Canon.html#LensType> whereas it is "Canon EF 14mm f/2.8L" in <source:trunk/src/canonmn.cpp#L834>.

Anything more to fix?

#2 - 25 Aug 2016 11:35 - Robin Mills

- Priority changed from Low to Normal

Lens recognition is tricky because every manufacturer does this a different way and it's very common for the LensID to be used by more than one lens. To recognise the lens, Exiv2 has code that inspects other metadata to "guess" the lens.

To enable the user to identify his lens, v0.26 (currently in development on the trunk) uses a file. On Linux/Mac/Cygwin/MinGW the file is ~/.exiv2 For Windows it's c:\Users\username\exiv2.ini. The file is stored in Windows .ini format. You can identify your lens with an entry such as:

```
[sony]
150=Sigma 18-35mm f/1.8
```

When you do this, all applications which use libexiv2 will recognise your lens. libexiv2 is used by GIMP, GwenView, digiKam and many other open source image processing tools.

This new feature is extensively discussed in [#1034http://dev.exiv2.org/issues/1034](http://dev.exiv2.org/issues/1034)

#3 - 25 Aug 2016 17:05 - Robin Mills

- % Done changed from 0 to 100
- Estimated time set to 1.00 h

#4 - 26 Aug 2016 00:00 - Martin Ramshaw

What has been proposed is only a work-around, not a fix.

It appears this has all but been marked 'closed' even so.

I applaud the attempt to enable savvy users to be able to shortcut the lengthy release cycle of this project. This issue of device ids, etc. is a pretty old one but this is hardly a high-priority project. USB ids for instance need a pretty rapid turn-around. Not carping, just a comment!

I'd suggest that the target be changed to 0.27 as well, as it looks like 0.26 is in 'code freeze' while you guys try to catch up to the announced release date.

Interestingly enough, I have had a lot of similar ideas myself in relation to LensFun; I only own a few lenses and I can make my own profiles (in fact, I'd prefer to do so) which I would like to apply in preference to the profiles supplied by LensFun. But I digress.

let me know what the projected 0.26 release date is, maybe I can test and submit a patch by then. Or not. As you like.

#5 - 26 Aug 2016 08:12 - Robin Mills

Martin

There is no fix for lens recognition when the manufacturer has used a LensID for more than one lens. When the lens is not unique, we have to inspect other metadata to deduce the lens in use. This is unsatisfactory. It is a guess.

The work-around by having the ~/.exiv2 file is a sensible feature. About 50% of issues reported to Exiv2 concern lens recognition and giving the user a way have his lens recognised saves everybody unhappiness. And it's instant. Nobody has to wait on a new release of software and the lens is known to every application using libexiv2.

The status of v0.26 is here and updated yesterday. This page is frequently updated: <http://dev.exiv2.org/news/2>

There is no point in assigning this to v0.27 as I don't want to spend further time on this. If however you wish to provide a test file and a code patch to correctly deduce your lens, I'll be happy to review and consider it for either v0.26 and v0.27.

#6 - 27 Aug 2016 00:50 - Martin Ramshaw

I had a feeling things were not going to be all that simple.

Command-line works, darktable (re-compiled) not so much:

```
kuby@Work:~/Documents/exiv2$ !1269
grep exiv2 exiv2.trace
execve("/usr/local/bin/exiv2", ["exiv2", "-pt", "IMG_0353.CR2"], [/* 67 vars */]) = 0
open("/usr/local/lib/libexiv2.so.14", O_RDONLY|O_CLOEXEC) = 3
open("/usr/local/share/locale/en_CA/LC_MESSAGES/exiv2.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/usr/local/share/locale/en/LC_MESSAGES/exiv2.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/usr/share/locale-langpack/en_CA/LC_MESSAGES/exiv2.mo", O_RDONLY) = 3
open("/usr/share/locale-langpack/en/LC_MESSAGES/exiv2.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/home/kuby/.exiv2", O_RDONLY) = 3
open("/home/kuby/.exiv2", O_RDONLY) = 3
kuby@Work:~/Documents/exiv2$ grep exiv2 darktable.trace
open("/opt/darktable/bin/./lib/darktable/libexiv2.so.14", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
open("/usr/local/lib/libexiv2.so.14", O_RDONLY|O_CLOEXEC) = 3
stat("/home/kuby/Documents/exiv2/IMG_0353.CR2", {st_mode=S_IFREG|0644, st_size=21815367, ...}) = 0
stat("/home/kuby/Documents/exiv2/IMG_0353.CR2", {st_mode=S_IFREG|0644, st_size=21815367, ...}) = 0
access("/home/kuby/Documents/exiv2/IMG_0353.CR2.xmp", F_OK) = 0
open("/home/kuby/Documents/exiv2/IMG_0353.CR2.xmp", O_RDONLY) = 29
stat("/home/kuby/Documents/exiv2/IMG_0353.CR2.xmp", {st_mode=S_IFREG|0664, st_size=2644, ...}) = 0
open("/home/kuby/Documents/exiv2/IMG_0353.CR2.xmp", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 29
kuby@Work:~/Documents/exiv2$ cat /.exiv2
```

[canon]

150--Sigma 18-35mm f/1.8 DC HSM [A]

```
kuby@Work:~/Documents/exiv2$ !1289
exiv2 -pt IMG_0353.CR2 | grep -ai lens
Exif.CanonCs.LensType          Short    1 Sigma 18-35mm f/1.8 DC HSM [A]
Exif.CanonCs.Lens              Short    3 18.0 - 35.0 mm
Exif.Canon.LensModel           Ascii    74 18-35mm
Exif.Photo.LensSpecification   Rational 4 18/1 35/1 0/1 0/1
```

```
Exif.Photo.LensModel          Ascii   8 18-35mm
Exif.Photo.LensSerialNumber   Ascii  11 0000000000
kuby@Work:~/Documents/exiv2$
```

As you can see from the traces above, darktable doesn't read ".exiv2".

Without knowing the details of the library code involved, this may or may not be correct, feedback is welcome.

#7 - 27 Aug 2016 01:22 - Martin Ramshaw

Okay, never mind, this appears to have been a darktable problem rather an exiv2 problem.

recompiled ufraw works as expected:

```
kuby@Work:~/Documents/exiv2/ufraw-0.22$ strace ufraw 2> ufraw.trace
kuby@Work:~/Documents/exiv2/ufraw-0.22$ grep exiv2 ufraw.trace
open("/usr/local/lib/libexiv2.so.14", O_RDONLY|O_CLOEXEC) = 3
stat("/home/kuby/Documents/exiv2/ufraw-0.22", {st_mode=S_IFDIR|0775, st_size=4096, ...}) = 0
open("/home/kuby/Documents/exiv2/IMG_0353.CR2", O_RDONLY) = 5
open("/home/kuby/Documents/exiv2/IMG_0353.CR2", O_RDONLY) = 6
stat("/home/kuby/Documents/exiv2/IMG_0353.CR2", {st_mode=S_IFREG|0644, st_size=21815367, ...}) = 0
stat("/home/kuby/Documents/exiv2/IMG_0353.CR2", {st_mode=S_IFREG|0644, st_size=21815367, ...}) = 0
stat("/home/kuby/Documents/exiv2/IMG_0353.CR2", {st_mode=S_IFREG|0644, st_size=21815367, ...}) = 0
open("/home/kuby/Documents/exiv2/IMG_0353.CR2", O_RDONLY) = 6
stat("/home/kuby/Documents/exiv2/IMG_0353.CR2", {st_mode=S_IFREG|0644, st_size=21815367, ...}) = 0
open("/home/kuby/Documents/exiv2/IMG_0353.CR2", O_RDONLY) = 6
stat("/home/kuby/Documents/exiv2/IMG_0353.CR2", {st_mode=S_IFREG|0644, st_size=21815367, ...}) = 0
stat("/home/kuby/Documents/exiv2/IMG_0353.CR2", {st_mode=S_IFREG|0644, st_size=21815367, ...}) = 0
stat("/home/kuby/Documents/exiv2/IMG_0353.CR2", {st_mode=S_IFREG|0644, st_size=21815367, ...}) = 0
open("/home/kuby/.exiv2", O_RDONLY) = 6
open("/home/kuby/.exiv2", O_RDONLY) = 6
open("/usr/local/share/locale/en_CA/LC_MESSAGES/exiv2.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/usr/local/share/locale/en/LC_MESSAGES/exiv2.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/usr/share/locale-langpack/en_CA/LC_MESSAGES/exiv2.mo", O_RDONLY) = 6
open("/usr/share/locale-langpack/en/LC_MESSAGES/exiv2.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
stat("/home/kuby/Documents/exiv2/ufraw-0.22", {st_mode=S_IFDIR|0775, st_size=4096, ...}) = 0
lstat("/home/kuby/Documents/exiv2/IMG_0353.png", 0x7ffdf4594b80) = -1 ENOENT (No such file or directory)
lstat("/home/kuby/Documents/exiv2/IMG_0353.png", 0x7ffdf4594750) = -1 ENOENT (No such file or directory)
lstat("/home/kuby/Documents/exiv2/IMG_0353.png", 0x7ffdf45920e0) = -1 ENOENT (No such file or directory)
getcwd("/home/kuby/Documents/exiv2/ufraw-0.22", 48) = 38
open("/home/kuby/Documents/exiv2/ufraw-0.22/.badpixels", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/home/kuby/Documents/exiv2/.badpixels", O_RDONLY) = -1 ENOENT (No such file or directory)
lstat("/home/kuby/Documents/exiv2/IMG_0353.png", 0x7ffdf4593dd0) = -1 ENOENT (No such file or directory)
kuby@Work:~/Documents/exiv2/ufraw-0.22$
```

#8 - 27 Aug 2016 12:49 - Robin Mills

Martin

You have to be sure that DT (or any application) is using a version of the library which contains the code to read ~/.exiv2. You can see libraries used by an application with the utility **lssof**. I've discussed this with somebody here: <http://dev.exiv2.org/issues/1181#note-2>

Please be aware that trunk builds of libexiv2 can cause problems with shipping binary versions of applications. To use trunk builds of libexiv2 with an application, I recommend totally rebuilding the application using exiv2 header files for that version of libexiv2. Rebuilding everything is a messy chore.

To avoid the burden of rebuilding applications, Pascal (he of DT fame) cherry picks the Exiv2 codebase and maintains libexiv2 v0.25 binaries which contain some v0.26 lens code. This is discussed here. <http://dev.exiv2.org/issues/1181#note-7>

#9 - 27 Aug 2016 19:16 - Martin Ramshaw

Not sure who Tim is, but you should let him know that when someone says they have recompiled an application it probably means just that.

Also, let him know that 'ldd' is more useful than 'lssof' for libraries:

```
kuby@Work:~$ ldd /opt/darktable/bin/darktable | grep exiv2
libexiv2.so.14 => /usr/local/lib/libexiv2.so.14 (0x00007fcd551f7000)
kuby@Work:~$ ldd /usr/bin/darktable | grep exiv2
libexiv2.so.12 => /usr/lib/libexiv2.so.12 (0x00007f2b30e82000)
kuby@Work:~$
```

Finally, let him know that if a library is in /usr/local/lib it is probably a WIP testing library that has been compiled there specifically.

[Also, when one is planning to fix some code, one really needs the source. Compiled binaries are not particularly useful for this purpose.]

#10 - 27 Aug 2016 19:36 - Robin Mills

Are we having a bad day, Martin? Apologies for calling you Tim.

I regret that I've said something amiss and you clearly did rebuild everything from source. It does however happen that folks recompile libexiv2 and encounter issues with binary distributions of applications. I see that you have understood that trap and done what's required to avoid it.

I'm not sure that ldd and lsof are alternatives. ldd reports information from an executable. lsof reports open files in memory. Both are useful in different ways.

#11 - 27 Aug 2016 20:39 - Robin Mills

- Status changed from Assigned to Closed

- Estimated time changed from 1.00 h to 2.00 h

#12 - 28 Aug 2016 00:48 - Martin Ramshaw

Well, the bug was assigned to Thomas. Tim **could** be short for Thomas.

Sorry if I came across as snarky, you didn't seem to have actually read the bug report so I felt compelled to prove that I know what I'm doing.

I think we can agree that the nice thing about *nix is a full set of tools. I have used 'lsof' in the past but it's definitely not high on my list.

I have been acquainted with Pascal for many years and am very grateful for his PPA and other work. However:

<http://www.darktable.org/2015/02/on-lens-detection-and-correction/>

In this article it seems that by getting a lens added to 'exiv2' that LensFun correction will then work.

Not so.

From my recent tests, 'ufraw' will do a fuzzy search of the LensFun database and will operate correctly - this does not depend on exiv2.

'darktable' will do a fuzzy search also but does not get it right. Not sure how or what DT is doing, but as 'ufraw' can work correctly it would seem to be a matter of replicating the 'ufraw' code in DT. I have had a look but got lost in the callbacks. Initially it was an idea of mine to try and find the libexiv2 API, but that seems unnecessary now.

Likewise 'gimplensfun' and 'digikam' (have not tested these two with the latest exiv2) do not correctly determine the lens either.

Things could be worse on my part: I could have spent time coming up with a patch to hardcode the various third-party lenses. But I expect this would have been fragile - if not unmaintainable - and not particularly welcome.

There are good reasons why third-party lens manufacturers will wish to re-use Canon lens ids, so this is a problem that will probably continue to get worse. However the exiv2.ini idea seems a good work-around.

Congratulations on avoiding the urge to over-engineer the exiv2.ini file. However I suspect that these will not be portable between *nix and Windows due to the '\r\n' versus '\n' issue. But I may be wrong.

#13 - 28 Aug 2016 07:08 - Robin Mills

Lots of useful information here, Martin. And thank you for your private email. I did read your notes #6 and #7 above, however I didn't understand your message. So I tried to keep you out of trouble by letting you know about related conversations.

You're right about the "nix" line-endings. The code uses library function **fgets(...)** to read lines from the .ini file. On the Mac, that handles UNIX (\n) and Windows (\r\n) and doesn't handle Mac (\r) lines. MSVC, Cygwin and Linux behave the same.

The src/ini.cpp code can use a local function instead of **fgets(...)**. I'm not going to do anything about this for v0.26 and probably never will. The "Mac Legacy \r" line-ending is a long buried relic. I've updated [#1034](#) about this and some other caveats concerning the format of that file.

