

Exiv2 - Feature #1034

Camera accessory overflow file

19 Feb 2015 08:44 - Tobias Jakobs

Status:	Closed	Start date:	19 Feb 2015
Priority:	Normal	Due date:	
Assignee:	Robin Mills	% Done:	100%
Category:	design	Estimated time:	30.00 hours
Target version:	0.26		
Description			
According to this forum post from Robin Mills: http://dev.exiv2.org/boards/3/topics/1766			
And the discussion here. http://www.darktable.org/2015/02/on-lens-detection-and-correction/comment-page-1/#comment-42822			
Provide an ascii file (XML, JSON, ...) for new cameras and accessories like lenses. This enable applications to update Maker Notes without compiling code.			
Related issues:			
Related to Exiv2 - Feature #1160: Feature Request - New Lens		Closed	03 Feb 2016
Related to Exiv2 - Feature #1170: Sigma 35mm F1.4 DG HSM on Canon 6D		Closed	22 Mar 2016
Related to Exiv2 - Feature #801: Add Lens "Sigma 18-200mm f/3.5-6.3 DC OS HSM..."		Closed	13 Jan 2012
Related to Exiv2 - Bug #867: Lens not detected on Olympus E-1		Assigned	10 Nov 2012
Related to Exiv2 - Bug #1200: Support for Sigma 17-70mm F2.8-4 DC Macro OS HS...		Closed	09 Aug 2016

Associated revisions

Revision 4249 - 25 Mar 2016 08:12 - Robin Mills

#1034 Added exiv2 config file support for lens recognition.

Revision 4250 - 25 Mar 2016 09:02 - Robin Mills

#1034. Corrections to r4249 to fix build-breakers

Revision 4251 - 25 Mar 2016 10:18 - Robin Mills

#1034. Corrections to r4249 to fix build-breakers

Revision 4253 - 25 Mar 2016 10:36 - Robin Mills

#1034. Corrections to r4249 to fix build-breakers

Revision 4254 - 25 Mar 2016 11:50 - Robin Mills

#1034. Corrections to r4249 to fix erroneous test suite messages.

Revision 4259 - 28 Mar 2016 13:33 - Robin Mills

#1034 Added msvc2005/ini-test

Revision 4293 - 29 Apr 2016 19:20 - Robin Mills

#1034 Added samples/toexv This is a first step to write exif metadata to memory to send to a web service.

Revision 4294 - 29 Apr 2016 19:24 - Robin Mills

#1034 Correction to r4293. Adding forgotten files!

Revision 4299 - 07 May 2016 16:40 - Robin Mills

#1034. Nikon support for ~/.exiv2

Revision 4300 - 07 May 2016 16:54 - Robin Mills

#1034 Pentax support for ~/.exiv2

Revision 4301 - 07 May 2016 18:44 - Robin Mills

#1034 Sony and Minolta support for ~/.exiv2

Revision 4320 - 12 Jun 2016 12:17 - Robin Mills

#1034 (and #1187). Fixed buildbreaker in MinGW/configure build.

Revision 4330 - 14 Jun 2016 21:25 - Robin Mills

#1034 and #867. Adding support for "Camera Overflow File" for olympus cameras.

Revision 4358 - 09 Aug 2016 20:55 - Robin Mills

#1200 and #1034. Adding support for ~/.exiv2 for Nikon Cameras.

History

#1 - 19 Feb 2015 11:10 - Thomas Beutlich

Actually the status quo is really annoying. Most times I svn update the exiv2 repo I can see makernote updates corresponding to camera/lense changes. Since exiv2 is released very rarely this means the hard-coded camera/lense data are out of sync most of the time exiv2 is used when it comes to new camera/lense models.

The enhanced feature actually is fairly simple to implement. One should decide on how to textual store the camera/lens data and how and when to read it from file. This is no big issue at all. One thing I have in mind is performance though. As a user of exiv2.lib (as opposed to exiv2.exe) (Yes, Win user I am.) I can easily live with reading the camera/lens data from file only once at the life time of the application. But having exiv2.exe as a stand-alone command-line tool reading this file with every start may decrease performance. So maybe one should think of splitting this feature for exiv2.lib and exiv2.exe.

#2 - 19 Feb 2015 14:28 - Robin Mills

- Target version set to 1.0

Gentlemen

We make formal release about once a year.

I agree that this feature is not very difficult to implement. We have XML support provided by expat, and the v0.25 library has JSON support provided by Jzon.cpp. The design has to be negotiated with Niels who is our makernote engineer. And the location of the file in the file system (or registry) has to be determined.

The performance issue is interesting. If we find the accessory in the "compiled in" tables, maybe we don't need to look in the overflow file. However there was a bug reported recently concerning two lenses having the same ID. We could not guess which was expected by the user. A possible fix for that is to provide the overflow file and search before checking the "compiled in" tables. The user could modify the overflow file to force his preference. If we adopted the "overflow first, compiled in second" design, we'd have to examine the performance implications.

All team members are volunteers and have busy lives with commitments to work/family/college/what-not.. If you wish to see a feature added to the library, you are welcome to join the team and write it. I think you'll discover that the effort to add features is more than you expect. It has to work on all supported platforms (Mac, Linux, Cygwin, MinGW and MSVC). You will need to add to our test suite which is written in bash. It needs to be documented. And you have to be willing to deal with user issues and bugs which arise.

I have thought of proposing this as a GSoC project. However, I don't have time in 2015 to be a mentor.

Robin

#3 - 19 Feb 2015 15:52 - Thomas Beutlich

Why 1.0? Milestone 1.0 translates to never (in my opinion as observer and part-time contributor of exiv2). Let's see what Niels suggests for the design and discuss it here.

#4 - 19 Feb 2015 17:04 - Robin Mills

Are you offering to undertake this project? I don't want to commit this to the v0.26 release and I haven't given a single thought to v0.27. So, I've declared it as v1.0.

#5 - 19 Feb 2015 20:37 - Niels Kristian Bech Jensen

A few thoughts about this. Not a real design specification, but more like a unstructured brain-storm. I think it can be developed in two stages:

1. I think such a system should be designed around XML. It is a mature and portable format (stage 1).
2. The text files should be considered temporary, meaning that they should be converted to source code and included in each release (stage 1).
3. The XML file(s) should supercede built in lens definition with same lens ID number to enable setting a "preferred lens" when several lenses share

the lens same ID number (stage 1).

4. The XML file(s) must be placed in a standard directory structure with priority to the users home directory. On Linux/Unix systems the search order should be: ~/.local/share/exiv2, /usr/local/share/exiv2, /usr/share/exiv2. Similar standards should be followed on other systems.

4. Scripts for automatic conversion of XML files to source code should be developed (stage 2). Perl or Python could be used for this and the next item.

5. Scripts for generation of XML files from ExifTool updates should be developed (stage 2).

Only camera models and lens models should be defined this way. All other metadata tags should be hardcoded for safety reasons.

Please comment on this and volunteer to implement it.

Best regards,
Niels Kristian Bech Jensen

#6 - 20 Feb 2015 12:16 - Thomas Beutlich

If I read it correctly you want to have the XML data on top of the hard-coded data and not instead, right?

#7 - 20 Feb 2015 13:24 - Robin Mills

- Status changed from New to Assigned

- Assignee set to Thomas Beutlich

- Target version changed from 1.0 to 0.26

I think we should use the strategy "overflow first, compiled-in second". However I don't think we can decide that without some performance testing.

I've assigned this issue to you and updated the Target to v0.26.

#8 - 20 Feb 2015 14:37 - Niels Kristian Bech Jensen

My idea is that the library should search the overflow files first and the hardcoded data second. For releases we will only include hardcoded data.

Best regards,
Niels Kristian Bech Jensen

#9 - 23 Feb 2015 03:34 - Andreas Huggel

And the discussion here.

<http://www.darktable.org/2015/02/on-lens-detection-and-correction/comment-page-1/#comment-42822>

Pascal's comment you're replying to is spot-on.

Re Niels' thoughts:

1. I think such a system should be designed around XML. It is a mature and portable format (stage 1).

Very much agree this should be limited to lenses. So if the format is simple, I'd prefer a simple text file that can be parsed without an XML parser. (To avoid the dependency on Expat - currently only needed for XMP - and the parsing overhead it introduces.)

Besides my main concerns are

a. Technical: This will require the library to perform some kind of one-time initialization, something which is ugly and hairy. We do have a precedent for that for XMP, but it's really not something I'd like to promote.

b. Conceptual: Chances are we won't get feedback from users anymore for lens definitions, they'll all just update their own files. So we'll rely on Exiftool completely. What if Phil adds a similar feature?

My idea is that the library should search the overflow files first and the hardcoded data second. For releases we will only include hardcoded data.

Agreed

#10 - 23 Feb 2015 09:47 - Tobias Jakobs

b. Conceptual: Chances are we won't get feedback from users anymore for lens definitions, they'll all just update their own files. So we'll rely on Exiftool completely.

This is of course pure guessing and I would expect the opposite. I expect more users to contribute new lenses, because they can test it directly and

benefit faster. I wrote the long version of this argument here:

<http://www.darktable.org/2015/02/on-lens-detection-and-correction/comment-page-1/#comment-42822>

At the moment I, as a user, have to wait ~1 year or longer to get benefit from a bugreport with a new lens. (New exiv2 version + new distribution version) I think in Exiftool I can very easily add a lens to the Perl file. (Not tested, my lens was already there.)

#11 - 23 Feb 2015 15:01 - Torsten Bronger

Is it at all possible to map exiv2's lens detection data to a not-too-complicated XML schema? As a related question, are we talking about moving the whole of exiv2's lens detection data into external text files, or only about infrastructure to add simple cases ad hoc?

#12 - 23 Feb 2015 16:02 - Niels Kristian Bech Jensen

It is meant as an ad hoc solution to add new lenses between releases.

Best regards,
Niels Kristian Bech Jensen

#13 - 25 Feb 2015 11:49 - Robin Mills

Here's another idea to consider for v0.25+.

We can create a new branch 'stable' which will begin life as a copy of v0.25. When any maker-note code is modified on 'trunk', the same code is synced on 'stable'. Anybody can download and build 'stable' during the life of v0.25 and get the latest makernote updates. The 'overflow' file isn't needed. And ... (drum roll) we can bring this to life as soon as v0.25 is published (taa-aaa). This doesn't conflict with adding the 'overflow' file feature in v0.26.

As we now have a Jenkins build server (<http://exiv2.dyndns.org:8080>) we can publish the latest binary build of 'stable' for download by anybody. For applications which use Exiv2 dynamic libraries, it's possible to 'patch' the application by downloading the built library.

This doesn't put a burden on Niels to keep trunk/makernotes -> stable/makernotes in perfect sync. There are only 11 makernote .cpp/.hpp file pairs and if we forget to sync a file occasionally, it will 'catch up' later. To relieve Niels of the sync burden, I can add a script to Jenkins to promote trunk/makernotes -> stable. To ensure quality, the build server builds and runs the test suite on all monitored branches.

```
638 rmills@rmillsmm:~/gnu/exiv2/foo/trunk $ finder "*" | rev | sort | rev
./src/sigmamn.cpp
./src/minoltamn.cpp
./src/panasonicmn.cpp
./src/samsungmn.cpp
./src/fujimn.cpp
./src/nikonmn.cpp
./src/canonmn.cpp
./src/casiomn.cpp
./src/olympusmn.cpp
./src/pentaxmn.cpp
./src/sonymn.cpp

./src/tiffmn-test.cpp

./src/sigmamn_int.hpp
./src/minoltamn_int.hpp
./src/panasonicmn_int.hpp
./src/samsungmn_int.hpp
./src/fujimn_int.hpp
./src/nikonmn_int.hpp
./src/canonmn_int.hpp
./src/casiomn_int.hpp
./src/olympusmn_int.hpp
./src/pentaxmn_int.hpp
./src/sonymn_int.hpp
639 rmills@rmillsmm:~/gnu/exiv2/foo/trunk $
```

Comments Welcome.

#14 - 23 Mar 2015 22:42 - Thomas Beutlich

Robin, does this really mean that stable binaries of library and command line tool are released (and provided by official distro repos like OpenSuse etc.) every time a maker note changes? Or does it mean that the source is provided and users can use and build them on their own?

#15 - 23 Mar 2015 23:15 - Robin Mills

Well, the long term plan is to add the 'overflow' file. However that's not going to be available until v0.26 which hopefully will ship in 2016. Niels has explained how the 'overflow' file will work after v0.26 ships.

v0.26 is quite a long way off. At the moment, I can't even say when v0.25 will ship. I don't have credentials for exiv2.org, so I have to wait for the

Exiv2 Release Engineers (Andreas and Shawn) to make that release.

However, I'm suggesting that v0.25 can have simple "dot releases" only involving makernote changes. We can have a very high confidence that the quality of v0.25.n = v0.25.0 if we only automatically sync changes to the above files into the "dot" branch.

This is just a suggestion. Thanks for asking for clarification. As nobody else has commented, I don't know if my suggestion is:

- Not understood
- Everybody's too busy to think about this
- Stupid and everybody's too polite to say so
- Something else!

However the intent would be that we would have lots of v0.25 "dot" releases. Any "dot" could be picked up by Suse etc. Every distro would have the latest makernote changes on the day they collect our code.

#16 - 27 Dec 2015 12:48 - Robin Mills

I've had a new idea about this while working on the camera lens issues such as [#1143](#) and [#1145](#).

We don't need an overflow file! We can have an SQL server on the internet and when an unidentified lens appears, libexiv2 can send a request using HTTP (which is part of all versions of libexiv2) along the lines of:

```
SELECT LensID where Exif.Image.Model = 'NIKON D5300' [AND key relop value]+"
```

By adopting such an architecture we:

1. Don't need an overflow file on the user's computer
2. We don't need dot releases with new lenses/accessories/cameras
3. Failing to identify a lens (or returning multiple lenses) can be logged so that the database notifies us of new devices

This solves the file format issue. We store the Lens database in an open-source database (such as MySQL). We might get lucky and be able to convert Phil's code and/or web site into SQL Tables.

This could be an interesting project. Perhaps we can get a Student from GSoC 2016 to implement this.

#17 - 28 Dec 2015 22:22 - Robin Mills

And here's another even simpler solution. exiftool understands .exv files (exiv2 extracted metadata files). If we don't know the lens, we can send the .exv file to an http server (such as exiv2.org) that has a CGI script to run exiftool on the .exv file. So simple. It's 100% cross platform because our http code is already cross platform. So we set this up and live happily ever after. We inherit all the wonderful work done by Phil and have no maintenance effort. No files on the client computer. No horrible dos box on Windows. No need for the client to install exiftool. And, we register failures and send the metadata to Phil - so he learns about new devices.

#18 - 02 Jan 2016 23:41 - Robin Mills

- Assignee changed from Thomas Beutlich to Robin Mills

- % Done changed from 0 to 10

- Estimated time set to 50.00 h

I'm going to take on this challenge. It's not very difficult. We have:

- 1) extract the .exv from the image and send it to the server
- 2) On the server, run exiftool on the .exv and transmit it to the client

To achieve this:

- 1) To transmit a .exv file to a server:

```
curl --silent --data-binary @Stonehenge.exv 'http://klanmills/exiftool.pl'
```

- 2) The cgi script to respond:

```
#!/usr/bin/perl

my $length = int $ENV{CONTENT_LENGTH};
my $exv;
read(STDIN, $exv, $length);

##
# $q is both the request and response!
use CGI qw(:standard);
my $q = CGI->new;
my @request = $q->http();

# are we a CGI script?
```

```

if ( scalar(@request) ) {
    ##
    # announce ourselves
    print $q->header(-expires=>'+0s') ;

    ##
    # save the payload
    my $filename = '/tmp/exiftool.exv';
    open(my $fh, '>', $filename) or die "Could not open file '$filename' $!";
    binmode($fh);
    syswrite $fh,$exv,$length;
    close $fh;

    ##
    # run exiftool on the payload
    print $q->header(-expires=>'+0s') ;
    system("/usr/local/bin/exiftool -j $filename");
}

# that's all Folks!
##

```

Obviously there are packaging and error handling matters to fix. However it doesn't look difficult.

Initially exiftool.pl can reside on clanmills.com or exiv2.dyndns.org. I'll provide an environment variable to redefine the server. Eg:

```
export EXIV2_EXIFTOOL=http://clanmills.com/exiftool.pl
```

I will ask Phil about using his code in this way. As the cgi script is written in Perl, there will be a more efficient way than system() to invoke exiftool.

#19 - 23 Mar 2016 09:00 - Robin Mills

I'm going to add the following to v0.26. The user can put the file .exiv2 (in .ini format) in their home directory to define a new lens:

```
[canon]
169=Sigma 35mm F1.4 DG HSM ART
```

Exiv2 will consult that file and use definitions in preference to what's compiled into the library. Let's not discuss the performance implication. I'll find a way to make it very fast (don't read if not necessary, then read/cache).

#20 - 25 Mar 2016 08:13 - Robin Mills

- % Done changed from 10 to 30

[r4249](#) Add 'overflow' file. exiv2 --verbose --version reports the path. Currently this is only used by canonmn.cpp.

The MSVC builds the path is HOMEDIR\exiv2.ini and ~/.exiv2 for others builds.

#21 - 07 May 2016 16:55 - Robin Mills

- % Done changed from 30 to 40

Added support ~/.exiv2 for Nikon and Pentax cameras

#22 - 07 May 2016 18:47 - Robin Mills

- % Done changed from 40 to 50

[r4301](#). Added sony and minolta support to ~/.exiv2

#23 - 14 Jun 2016 21:32 - Robin Mills

- Status changed from Assigned to Closed

- % Done changed from 50 to 100

- Estimated time changed from 50.00 h to 30.00 h

[r4330](#) Added olympus support to ~/.exiv2

Support for "camera overflow file" has been added in v0.26 for the following cameras:

1. Canon
2. Minolta (Sony)

3. Nikon
4. Olympus
5. Pentax

I'm reducing the scope of this feature for v0.26. The web-service feature has been partially developed and documented here, however I'm not going to develop it for v0.26. I'm intending to get a **GSoC** student in 2017 to work on [#992](#) Raw Image Support. Let's wait to see how that develops to see if the exiftool web service is required after that work has been completed.

#24 - 28 Aug 2016 07:07 - Robin Mills

In a conversation with Martin, he raised the subject of line-endings in ~/.exiv2 <http://dev.exiv2.org/issues/1212#note-13>

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 code in 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. In the process of testing line endings on Linux, I consulted the man page for unix2dos and unix2mac. Gosh, there's a mine-field of ways to complicate this file including character sets, BOM markers and all manner of stuff. This feature works well enough on "simple/plain" Unix and DOS files.