

## Exiv2 - Bug #1289

### Infinite loop on command exiv2 -pR on a .CR2 file

14 Apr 2017 19:42 - Robin Mills

<b>Status:</b>	Closed	<b>Start date:</b>	14 Apr 2017
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Ben Touchette	<b>% Done:</b>	100%
<b>Category:</b>	image format	<b>Estimated time:</b>	3.00 hours
<b>Target version:</b>	0.26		
<b>Description</b>			
This has been reported by Asdiel (AlienSkin Software). Test file at <a href="http://exiv2.dyndns.org:8080/userContent/testfiles/1289/IMG_1538.CR2">http://exiv2.dyndns.org:8080/userContent/testfiles/1289/IMG_1538.CR2</a>			

#### Associated revisions

##### Revision 4754 - 14 Apr 2017 19:42 - Robin Mills

#1289 Thanks to Asdiel (AlienSkin) for reporting this and providing a test file. Thanks to Ben for investigation and patch.

#### History

##### #1 - 14 Apr 2017 19:46 - Robin Mills

- Category set to image format
- Assignee set to Ben Touchette
- Target version set to 0.26
- % Done changed from 0 to 100
- Estimated time set to 3.00 h

Fix submitted [r4755](#). Thank You to Ben for providing the patch.

Ben: I shortened your fix by 4 lines, however it is your fix. Thanks for working on this.

##### #2 - 14 Apr 2017 20:10 - Robin Mills

A couple of additional observations about this:

- 1) Why is this looping when TiffVisitor reads it successfully?  
I should investigate to discover why printStructure() dies on this file and TiffVisitor does not.
- 2) Why are we frequently running exiv2 -pR for no good reason?

Asdiel has pointed out that we shouldn't be executing **Image::printStructure(kpsRecursive)** on files which are not remote. This code is only required by Remotelo objects to ensure that the metadata is in memory for TiffVisitor.

I have proposed method Basiclo::isRemote() which would return true for remote data sources. We could add the following to Image::printIFDStructure() after the first write:

```
if ( !out.good() && !io->isRemote() ) return;
```

This says skip printIFDStructure() on a stream attached to /dev/null when the IO source is remote.

It's easy to define isRemote() for files using protocols such as http/ftp/ssh. However local files which are network shares/mounts (afp, nfs, samba) require investigation.

##### #3 - 15 Apr 2017 21:19 - Robin Mills

I've investigated the subject of how to determine if a file is on local storage or a network share. On Windows there is an API GetDeviceType(path) which looks like our man! [https://msdn.microsoft.com/en-us/library/windows/desktop/aa364939\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa364939(v=vs.85).aspx)

On Mac/Linux, one solution is to call realpath (*see man 3 realpath*). If realpath starts with /media (*Linux*) or /Volumes (*MacOS-X*), it's a network share.

So the default implementation of Basiclo::isRemote() is to return false. For class Remotelo (and derived classes), isRemote() returns true. Filelo::isRemote can be implemented on Windows (including MinGW and Cygwin) using GetDeviceType. Linux and MacOS-X can use the realpath /media or /Volumes strategy.

#4 - 15 Apr 2017 21:36 - Robin Mills

I've investigated why the test file loops on Image::printIFDStructure and survives TiffVisitor. I don't know the exact reason, however Ben's fix is very effective. We stop processing the subIFD in the .CR2.

```
778 rmills@rmillsmbp:~/Downloads $ exiv2 -pR IMG_1538.CR2 2>&1
```

```

...
STRUCTURE OF TIFF FILE (II): IMG_1538.CR2
address | tag | type | count | offset | value
 998 | 0x0001 | Version | SHORT | 49 | 1518 | 98 2 0 4 0 ...
1010 | 0x0002 | ISOSpeed | SHORT | 4 | 1616 | 0 85 63343 21770
1022 | 0x0003 | ColorMode | SHORT | 4 | 1624 | 0 0 0 0
1034 | 0x0004 | Quality | SHORT | 34 | 1632 | 68 0 160 52 52 ...
1046 | 0x0006 | Sharpening | ASCII | 20 | 1700 | Canon EOS REBEL SL1
1058 | 0x0007 | Focus | ASCII | 24 | 1732 | Firmware Version 1.0.0.
1070 | 0x0009 | GPSStatus | ASCII | 32 | 1756 | .....
.....
1082 | 0x000d | GPSSpeed | UNDEFINED | 1536 | 1788 | .._`.H..../.....
.....T ...
1094 | 0x0010 | DataDump | LONG | 1 | 2147484486 | 2147484486
1106 | 0x0013 | GPSDestLatitudeRef | SHORT | 4 | 3324 | 0 159 7 112
1118 | 0x0019 | GPSDestDistanceRef | SHORT | 1 | 1 | 1
1130 | 0x0026 | | SHORT | 139 | 3332 | 278 2 31 9 5184 ...
1142 | 0x0035 | | LONG | 4 | 3610 | 16 4294966816 30 0
1154 | 0x0093 | | SHORT | 32 | 3626 | 64 0 0 0 0 ...
1166 | 0x0095 | | ASCII | 74 | 3690 | EF85mm f/1.8 USM.....
.....
1178 | 0x0096 | | ASCII | 16 | 3764 | HA0751226.....
1190 | 0x0097 | | UNDEFINED | 1024 | 3780 | .....
.....
1202 | 0x0098 | | SHORT | 4 | 4804 | 0 0 0 0
1214 | 0x0099 | | LONG | 38 | 4812 | 152 4 1 32 2 ...
1226 | 0x009a | | LONG | 5 | 4964 | 0 5184 3456 0 0
1238 | 0x00a0 | | SHORT | 14 | 4984 | 28 0 3 0 0 ...
1250 | 0x00aa | | SHORT | 6 | 5012 | 12 835 1024 1024 407 ...
1262 | 0x00b4 | | SHORT | 1 | 1 | 1
1274 | 0x00d0 | | LONG | 1 | 0 | 0
1286 | 0x00e0 | | SHORT | 17 | 5024 | 34 5280 3528 1 1 ...
1298 | 0x4001 | | SHORT | 1313 | 5058 | 10 749 1024 1024 348 ...
1310 | 0x4002 | | UNDEFINED | 43636 | 7684 | t..0".....
.....
1322 | 0x4005 | | UNDEFINED | 16796 | 51320 | .A.....
.....
1334 | 0x4008 | | SHORT | 3 | 68116 | 135 135 135
1346 | 0x4009 | | SHORT | 3 | 68122 | 0 0 0
1358 | 0x4010 | | ASCII | 32 | 68128 | .....
.....
1370 | 0x4011 | | UNDEFINED | 252 | 68160 | .....
.....
1382 | 0x4012 | | ASCII | 32 | 68412 | .....
.....
1394 | 0x4013 | | LONG | 11 | 68444 | 44 0 0 10 4294967295 ...
1406 | 0x4015 | | UNDEFINED | 456 | 68488 | .!.....@..
.....
1418 | 0x4016 | | LONG | 7 | 68944 | 28 0 1 0 1 ...
1430 | 0x4018 | | LONG | 7 | 68972 | 28 0 0 0 0 ...
1442 | 0x4019 | | UNDEFINED | 30 | 69000 | .....
.....
1454 | 0x4020 | | LONG | 7 | 69030 | 28 0 0 0 2147483647 ...
invalid type value detected in Image::printIFDStructure: 0
END IMG_1538.CR2

```

I suspect the TiffVisitor behaves in a similar manner and quietly stops processing this subIFD. TiffVisitor does detect and warn:

```
Warning: Directory Canon, entry 0x0000 has unknown Exif (TIFF) type 0; setting type size 1.
```

I'm going to leave this issue closed. I'm unwilling at this time in v0.26 to introduce the new API Basiclo::isRemote() and the implementation. It's not a lot of work, however it's not essential. And with the C++ implementation being platform dependent, there will probably be build and test matters which will arise from this. And finally, there is no question that the file has illegal data.