

Exiv2 - Bug #677

Nikon Makernote tags regression in exiv 0.19

27 Jan 2010 13:44 - Mark Purcell

Status: Closed	Start date: 27 Jan 2010
Priority: Normal	Due date:
Assignee:	% Done: 100%
Category: metadata	Estimated time: 0.00 hour
Target version: 0.20	

Description

https://bugs.kde.org/show_bug.cgi?id=224094

----- Comment #19 From Mark Purcell 2010-01-27 22:26:29 () [reply] -----
Using the test file from #6 & exiv2 0.19 it takes some 20 seconds to reset
NoName:

```
$ time exiv2 -M'set Exif.Image.Make NoName' digikam-example-d90.jpg
```

```
real 0m20.371s
user 0m20.141s
sys 0m0.052s
```

Downgrading to exiv2 0.18.2 (and refreshing the test file) works much smoother:

```
mark@hp:~/Desktop/digikam-test$ exiv2 -V
exiv2 0.18.2
Copyright (C) 2004-2009 Andreas Huggel.
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

```
$ time exiv2 -v -M'set Exif.Image.Make NoName' digikam-example-d90.jpg
File 1/1: digikam-example-d90.jpg
Set Exif.Image.Make "NoName" (Ascii)
```

```
real 0m0.087s
user 0m0.020s
sys 0m0.032s
```

So issue isn't with digikam, rather exiv2..

I have placed a copy of exiv2 0.18.2 .deb at <http://people.debian.org/~msp/>, however this will still require the rebuild of libkexiv2, which is non-trivial..

Related issues:

Related to Exiv2 - Feature #666: Optimize binary array elements	Closed	29 Dec 2009
Related to Exiv2 - Feature #620: Update Nikon makernotes	Closed	14 Mar 2009

Associated revisions

Revision 2017 - 30 Jan 2010 01:50 - Andreas Huggel

#677: Changed the container for Exif metadata from a `std::vector` to a `std::list` (plus an unrelated optimization).

Revision 2018 - 30 Jan 2010 02:05 - Andreas Huggel

#677: Avoid deleting Exif metadata from the container when writing.

Revision 2019 - 30 Jan 2010 22:31 - Andreas Huggel

#677: Reversed changes from r2018.

Revision 2020 - 31 Jan 2010 05:03 - Andreas Huggel

Various optimizations (side-effect of the analysis for #677).

History

#1 - 27 Jan 2010 22:32 - Andreas Huggel

Because of the way the Nikon Makernote currently works, the Exif metadata of this image has 6785 tags (mostly unknown, see [#666](#)). An average image with a different Makernote probably has only in the order of 100 Exif tags.

A quick analysis with `kcachegrind` shows the problematic area is this sequence of calls:

```
. TiffEncoder::encodeTiffComponent (called 13530 times)
.. ExifData::erase (6769)
... std::vector<Exifdatum>::erase (6769)
.... Exifdatum::operator= (called 22.9M times!!!)
```

This happens during intrusive writing. Erasing elements of a `std::vector` apparently becomes a major issue with the larger number of elements and a non-trivial assignment operator.

What can be done?

- Use a different std container with a cheaper erase method
- Avoid deleting the elements from the container
- Reduce the number of Exif tags decoded from Nikon Makernotes (by implementing [#666](#))
- Improve performance of the assignment operator (e.g., by using a smart pointer with a copy-on-write strategy for the data members of `Exifdatum`)

#2 - 30 Jan 2010 00:03 - Andreas Huggel

Current status, [r2016](#):

```
andreas@mowgli:~/src/exiv2/trunk/src$ time ./exiv2 -M'set Exif.Image.Make NoName' digikam-example-d90.jpg

real    0m22.555s
user    0m20.065s
sys     0m0.200s
andreas@mowgli:~/src/exiv2/trunk/src$ cp ../../pic/digikam-example-d90.jpg .
cp: overwrite './digikam-example-d90.jpg'? y
andreas@mowgli:~/src/exiv2/trunk/src$ valgrind ./exiv2 -M'set Exif.Image.Make NoName' digikam-example-d90.jpg
==25441== Memcheck, a memory error detector
==25441== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
==25441== Using Valgrind-3.5.0-Debian and LibVEX; rerun with -h for copyright info
==25441== Command: ./exiv2 -Mset\ Exif.Image.Make\ NoName digikam-example-d90.jpg
==25441==
==25441== HEAP SUMMARY:
==25441==    in use at exit: 0 bytes in 0 blocks
==25441==   total heap usage: 69,529,350 allocs, 69,529,350 frees, 1,219,947,195 bytes allocated
==25441==
==25441== All heap blocks were freed -- no leaks are possible
==25441==
==25441== For counts of detected and suppressed errors, rerun with: -v
==25441== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 21 from 6)
```

Crazy... and a nice optimization challenge

#3 - 30 Jan 2010 01:50 - Andreas Huggel

[r2017](#): Changed the std container from a `std::vector` to a `std::list` (plus an unrelated optimization). A small (4 lines) and simple change with a drastic result:

```

andreas@mowgli:~/src/exiv2/trunk/src$ cp ../../pic/digikam-example-d90.jpg .
cp: overwrite './digikam-example-d90.jpg'? y
andreas@mowgli:~/src/exiv2/trunk/src$ time ./exiv2 -M'set Exif.Image.Make NoName' digikam-example-d90.jpg

real    0m0.486s
user    0m0.464s
sys     0m0.016s
andreas@mowgli:~/src/exiv2/trunk/src$ cp ../../pic/digikam-example-d90.jpg .
cp: overwrite './digikam-example-d90.jpg'? y
andreas@mowgli:~/src/exiv2/trunk/src$ valgrind ./exiv2 -M'set Exif.Image.Make NoName' digikam-example-d90.jpg
==30722== Memcheck, a memory error detector
==30722== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
==30722== Using Valgrind-3.5.0-Debian and LibVEX; rerun with -h for copyright info
==30722== Command: ./exiv2 -Mset\ Exif.Image.Make\ NoName digikam-example-d90.jpg
==30722==
==30722==
==30722== HEAP SUMMARY:
==30722==   in use at exit: 0 bytes in 0 blocks
==30722== total heap usage: 353,962 allocs, 353,962 frees, 22,674,838 bytes allocated
==30722==
==30722== All heap blocks were freed -- no leaks are possible
==30722==
==30722== For counts of detected and suppressed errors, rerun with: -v
==30722== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 21 from 6)

```

#4 - 30 Jan 2010 21:24 - Andreas Huggel

[r2018](#): Don't delete Exif metadata during writing.

```

andreas@mowgli:~/src/exiv2/trunk/src$ time ./exiv2 -M'set Exif.Image.Make NoName' digikam-example-d90.jpg

real    0m4.604s
user    0m4.480s
sys     0m0.016s
andreas@mowgli:~/src/exiv2/trunk/src$ cp ../../pic/digikam-example-d90.jpg .
cp: overwrite './digikam-example-d90.jpg'? y
andreas@mowgli:~/src/exiv2/trunk/src$ valgrind ./exiv2 -M'set Exif.Image.Make NoName' digikam-example-d90.jpg
==6819== Memcheck, a memory error detector
==6819== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
==6819== Using Valgrind-3.5.0-Debian and LibVEX; rerun with -h for copyright info
==6819== Command: ./exiv2 -Mset\ Exif.Image.Make\ NoName digikam-example-d90.jpg
==6819==
==6819==
==6819== HEAP SUMMARY:
==6819==   in use at exit: 0 bytes in 0 blocks
==6819== total heap usage: 353,962 allocs, 353,962 frees, 22,674,837 bytes allocated
==6819==
==6819== All heap blocks were freed -- no leaks are possible
==6819==
==6819== For counts of detected and suppressed errors, rerun with: -v
==6819== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 21 from 6)

```

Interestingly, now that deleting an element from the container is cheap, not deleting them makes things worse. That's because `ExifData::findKey()` has to work much harder if all the elements remain in the container, compared to before when every find operation worked on a container with one element less. I'll undo this change.

#5 - 30 Jan 2010 22:28 - Andreas Huggel

- File `callgrind.out.r2017` added
- File `callgrind.out.r2018` added

#6 - 31 Jan 2010 05:36 - Andreas Huggel

- Status changed from *New* to *Resolved*
- % Done changed from 0 to 100

Changed the container for Exif metadata to a `std::list` plus a few minor optimizations. With that, performance is acceptable again, taking into account the large number of tags in the Makernote of this image. That will be reduced considerably with the fix for [#666](#), after which images with Nikon makernotes will again perform like any other.

#7 - 05 Feb 2010 18:35 - Andreas Huggel

With the fix for [#666](#) the number of tags decoded from the sample image is reduced to only 206 (by concatenating unknown tags):

```

andreas@mowgli:~/src/exiv2/trunk/src$ time ./exiv2 -M'set Exif.Image.Make NoName' digikam-example-d90.jpg

real    0m0.031s
user    0m0.020s
sys     0m0.008s
andreas@mowgli:~/src/exiv2/trunk/src$ \cp -f ../../pic/digikam-example-d90.jpg .
andreas@mowgli:~/src/exiv2/trunk/src$ valgrind ./exiv2 -M'set Exif.Image.Make NoName' digikam-example-d90.jpg
==5057== Memcheck, a memory error detector
==5057== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
==5057== Using Valgrind-3.5.0-Debian and LibVEX; rerun with -h for copyright info
==5057== Command: ./exiv2 -Mset\ Exif.Image.Make\ NoName digikam-example-d90.jpg
==5057==
==5057==
==5057== HEAP SUMMARY:
==5057==     in use at exit: 0 bytes in 0 blocks
==5057==   total heap usage: 11,821 allocs, 11,821 frees, 2,138,217 bytes allocated
==5057==
==5057== All heap blocks were freed -- no leaks are possible
==5057==
==5057== For counts of detected and suppressed errors, rerun with: -v
==5057== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 21 from 6)

```

#8 - 05 Feb 2010 18:46 - Andreas Huggel

And finally, for comparison, the same tests with 0.18.2, which decodes 131 tags from the sample image (on the same machine, also compiled with debug symbols):

```

andreas@mowgli:~/src/exiv2/exiv2-0.18.2/src$ time ./exiv2 -M'set Exif.Image.Make NoName' digikam-example-d90.j
pg
real    0m0.035s
user    0m0.012s
sys     0m0.016s
andreas@mowgli:~/src/exiv2/exiv2-0.18.2/src$ \cp -f ../../pic/digikam-example-d90.jpg .
andreas@mowgli:~/src/exiv2/exiv2-0.18.2/src$ valgrind ./exiv2 -M'set Exif.Image.Make NoName' digikam-example-d
90.jpg
==13070== Memcheck, a memory error detector
==13070== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
==13070== Using Valgrind-3.5.0-Debian and LibVEX; rerun with -h for copyright info
==13070== Command: ./exiv2 -Mset\ Exif.Image.Make\ NoName digikam-example-d90.jpg
==13070==
==13070==
==13070== HEAP SUMMARY:
==13070==     in use at exit: 0 bytes in 0 blocks
==13070==   total heap usage: 29,825 allocs, 29,825 frees, 5,454,252 bytes allocated
==13070==
==13070== All heap blocks were freed -- no leaks are possible
==13070==
==13070== For counts of detected and suppressed errors, rerun with: -v
==13070== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 21 from 6)

```

#9 - 29 May 2010 10:43 - Andreas Huggel

- Status changed from Resolved to Closed

Files

File Name	Size	Date	Author
digikam-example-d90.jpg	1.96 MB	27 Jan 2010	Mark Purcell
callgrind.out.r2017	1.35 MB	30 Jan 2010	Andreas Huggel
callgrind.out.r2018	1.35 MB	30 Jan 2010	Andreas Huggel