Exiv2 - Bug #1187

Crash while reading in parallel threads

29 May 2016 08:06 - Taras Kushnir

Status:	Closed	**Start date:**	29 May 2016
Priority:	Normal	**Due date:**	
Assignee:	Robin Mills	**% Done:**	100%
Category:	xmp	**Estimated time:**	15.00 hours
Target version:	0.26		

Description

I'm using exiv2 library for my project, Xpiks (http://ribtoks.github.io/xpiks/). I've implemented parallel reading of metadata, but it seem to be crashing every second time.

Operating system: OS X 10.10.5
Exiv2 version: 0.25 - release

I really need a patch for this asap, since the bug started to repro suddenly and I've already released my version with exiv2 (previously I used exiftool).

Full stacktraces here:
http://pastebin.com/egUDkf3X

Code working with exiv2 is here:
https://github.com/Ribtoks/xpiks/blob/master/src/xpiks-qt/MetadataIO/exiv2readingworker.cpp (from the line 552)
Threading here:
https://github.com/Ribtoks/xpiks/blob/master/src/xpiks-qt/MetadataIO/readingorchestrator.cpp (nothing unusual)

Stacktraces related to exiv2:

```
Thread 25 Crashed:: QThread
0   com.yourcompany.xpiks-qt        0x0000000101865c9d std::__1::__tree_node_base<void*>* std::__1::__tree_next<std::__1::__tree_node_base<void*>*>(std::__1::__tree_node_base<void*>*) + 109
1   com.yourcompany.xpiks-qt        0x00000001017b8c60 Exiv2::XmpProperties::lookupNsRegistry(Exiv2::XmpNsInfo::Prefix const&) + 1136
2   com.yourcompany.xpiks-qt        0x00000001017bafd5 Exiv2::XmpProperties::ns(std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&) + 69
3   com.yourcompany.xpiks-qt        0x00000001017bc958 Exiv2::XmpKey::Impl::Impl(std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&, std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&) + 440
4   com.yourcompany.xpiks-qt        0x00000001017bcb55 Exiv2::XmpKey::Impl::Impl(std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&, std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&) + 37
5   com.yourcompany.xpiks-qt        0x00000001017bd4d8 Exiv2::XmpKey::XmpKey(std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&, std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&) + 104
6   com.yourcompany.xpiks-qt        0x00000001017bd555 Exiv2::XmpKey::XmpKey(std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&, std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&) + 37
7   com.yourcompany.xpiks-qt        0x000000010188fdd4 Exiv2::XmpParser::decode(Exiv2::XmpData&, std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&) + 15220
8   com.yourcompany.xpiks-qt        0x000000010188d3a9 Exiv2::XmpParser::decode(Exiv2::XmpData&, std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&) + 4425
9   com.yourcompany.xpiks-qt        0x0000000101734d61 Exiv2::JpegBase::readMetadata() + 3889
10  com.yourcompany.xpiks-qt        0x00000001015652b2 MetadataIO::Exiv2ReadingWorker::readMetadata(Models::ArtworkMetadata*, MetadataIO::ImportDataResult&) + 178
11  com.yourcompany.xpiks-qt        0x0000000101564866 MetadataIO::Exiv2ReadingWorker::process() + 470
12  org.qt-project.QtCore           0x00000001028202e2 QMetaObject::activate(QObject*, int, int, void**) + 2994
13  org.qt-project.QtCore           0x0000000102644d76 QThreadStorageData::finish(void**) + 2662
14  libsystem_pthread.dylib         0x00007fff88d4705a _pthread_body + 131
15  libsystem_pthread.dylib         0x00007fff88d46fd7 _pthread_start + 176
16  libsystem_pthread.dylib         0x00007fff88d443ed thread_start + 13
```

```
Thread 26:: QThread
0   libsystem_kernel.dylib          0x00007fff8b79d166 _psynch_mutexwait + 10
1   com.yourcompany.xpiks-qt          0x000000010162ffa5 XMP_EnterCriticalRegion(_opaque_pthread_mutex_t&) + 21
2   com.yourcompany.xpiks-qt          0x0000000101661f02 XMP_AutoMutex::XMP_AutoMutex() + 34
3   com.yourcompany.xpiks-qt          0x0000000101661e95 XMP_AutoMutex::XMP_AutoMutex() + 21
4   com.yourcompany.xpiks-qt          0x00000001016618b4 WXMPIterator_Next_1 + 68
5   com.yourcompany.xpiks-qt          0x000000010189901c TXMPIterator<std::_1::basic_string<char, std::__1::char_traits<char>,
std::__1::allocator<char> > >::Next(std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> >,
std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> >, std::__1::basic_string<char,
std::__1::char_traits<char>, std::__1::allocator<char> >, unsigned long) + 188
6   com.yourcompany.xpiks-qt          0x000000010188e339 Exiv2::XmpParser::decode(Exiv2::XmpData&,
std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&) + 8409
7   com.yourcompany.xpiks-qt          0x0000000101734d61 Exiv2::JpegBase::readMetadata() + 3889
8   com.yourcompany.xpiks-qt          0x00000001015652b2
MetadataIO::Exiv2ReadingWorker::readMetadata(Models::ArtworkMetadata*, MetadataIO::ImportDataResult&) + 178
9   com.yourcompany.xpiks-qt          0x0000000101564866 MetadataIO::Exiv2ReadingWorker::process() + 470
10  org.qt-project.QtCore             0x00000001028202e2 QMetaObject::activate(QObject*, int, int, void**) + 2994
11  org.qt-project.QtCore             0x0000000102644d76 QThreadStorageData::finish(void**) + 2662
12  libsystem_pthread.dylib           0x00007fff88d4705a _pthread_body + 131
13  libsystem_pthread.dylib           0x00007fff88d46fd7 _pthread_start + 176
14  libsystem_pthread.dylib           0x00007fff88d443ed thread_start + 13

Thread 25 crashed with X86 Thread State (64-bit):
rax: 0x0000000101be3da8  rbx: 0x000060800056b580  rcx: 0x0000000000000000  rdx: 0x0000000000000001
rdi: 0x00006080000bdac0  rsi: 0x0000000000000000  rbp: 0x000000013adaafb0  rsp: 0x000000013adaafb0
r8: 0x656c6261746b7261   r9: 0x0000000000000000  r10: 0xffffffffffffffff  r11: 0xffff9f813ab98651
r12: 0x00007ffed3dd5560  r13: 0x000000013adadc80  r14: 0x000000013adadc80  r15: 0x0000608000244f50
rip: 0x0000000101865c9d  rfl: 0x0000000000010202  cr2: 0x0000000000000000
```

Related issues:

Related to Exiv2 - Feature #1188: Provide build support for C++11	**Closed**	**29 May 2016**
Related to Exiv2 - Bug #1207: digiKam maintenance tool to synchronize files m...	**Closed**	**13 Aug 2016**
Related to Exiv2 - Bug #1270: Using libexiv2.a/.lib in multhreaded app segfa...	**Closed**	**13 Jan 2017**

Associated revisions

Revision 4308 - 30 May 2016 14:45 - Robin Mills

#1187 Thank You to Taras for the patch.

Revision 4309 - 01 Jun 2016 20:17 - Robin Mills

#1187 Fixing msvc build breaker in r4308 Thank You Taras for the patch.

Revision 4311 - 04 Jun 2016 05:56 - Robin Mills

#1187 and #1041 Fixing CMake/MSVC build breaker

Revision 4312 - 04 Jun 2016 17:08 - Robin Mills

#1187 and #1041. Fixing CMake/MSVC 2013/15 build breakers

Revision 4313 - 05 Jun 2016 06:15 - Robin Mills

#1187 and #1041. Fixing CMake/Linux build breakers concerning libpthread

Revision 4314 - 07 Jun 2016 15:27 - Robin Mills

#1187 Fixing ./configure build breaker concerning libpthread

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

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

Revision 4323 - 13 Jun 2016 13:18 - Robin Mills

#1187 and #1041. Fixing CMake to install include/exiv2/rwlock.hpp (detected by contrib/buildserver/test_daily.sh)

Revision 4324 - 13 Jun 2016 15:14 - Robin Mills

#1187 and #1041. Fixing src/Makefile to install include/exiv2/rwlock.hpp (detected by contrib/buildserver/test_daily.sh)

History

#1 - 29 May 2016 08:14 - Taras Kushnir

It repros only in Release build for me so I can't debug it.

#2 - 29 May 2016 08:40 - Gilles Caulier

digiKam use multithreading to do exactly the same task than you, and it do not crash. We use also Qt as i'm sure that you know already as i i see my comments in your code.

Your trace show a XMP problem. 99% of chance that XMP SDK from Adobe is not initiate properly at your application startup...

Gilles Caulier

#3 - 29 May 2016 08:47 - Taras Kushnir

Gilles Caulier wrote:

> Your trace show a XMP problem. 99% of chance that XMP SDK from Adobe is not initiate properly at your application startup...

Thanks for the fast response!

Umm. Can you please tell me how to initiate it "properly"?

#4 - 29 May 2016 09:11 - Robin Mills

- Category set to xmp

- Assignee set to Robin Mills

- Priority changed from Urgent to Normal

- Target version set to 0.26

I made a change last September in lookupNsRegistry() and I'm wondering if this is thread-safe. I'm on vacation at the moment and my time to investigate is limited. Are you using the shipping version of v0.25 which does not have my change to lookupNsRegistry()?

#5 - 29 May 2016 09:17 - Taras Kushnir

Robin Mills wrote:

> I made a change last September in lookupNsRegistry() and I'm wondering if this is thread-safe. I'm on vacation at the moment and my time to investigate is limited. Are you using the shipping version of v0.25 which does not have my change to lookupNsRegistry().

I'm using "latest stable" I was able to find which is v0.25 atm.

A have added a call to Exiv2::XmpParser::initialize() in the beginning of my app's startup and it still crashes inside lookupNsRegistry().

Overall I didn't see how 0.26 is going, but static variables and threading is a sure recipe for disaster.

BTW, is trunk in SVN "production ready"? Or there're is some big testing planned?

#6 - 29 May 2016 10:37 - Taras Kushnir

This issue has nothing to do with initialize() which is racy as well (if you don't call it in the beginning of the startup).

The problem is that XmpProperties::registerNs(), XmpProperties::lookupNsRegistry() and XmpProperties::unregisterNs()
have pretty obvious race conditions accessing std::map<std::string, XmpNsInfo> _nsRegistry.

All of them should have locking/etc. Code in trunk (is it future 0.26) is same as 0.25 for the methods above.

#7 - 29 May 2016 12:55 - Taras Kushnir

I've created a patch with a fix and tested it running ~150 threads in parallel. It crashes if I use old lib and does not if I use lib with my patch. I've added RW lock for operations with NsRegistry nsRegistry_

So the patch for:

- properties.cpp
https://github.com/Ribtoks/xpiks/blob/db25d22f7df71ba3b18bbf5e57a2d94658ddf44b/src/exiv2-0.25/rwlock.patch

- properties.h
is topmost here
https://github.com/Ribtoks/xpiks/commit/db25d22f7df71ba3b18bbf5e57a2d94658ddf44b

- new file rwlock.hpp
https://github.com/Ribtoks/xpiks/blob/77e356af7b02a5a6e9aecb4c1f2966a79c618d79/src/exiv2-0.25/include/exiv2/rwlock.hpp

I know every project has it's own rules for submitting patches and stuff, but I'm really super busy right now to read them and read docs about SVN how to submit my patch over there
(it would be so much easier if you guys just used GitHub or Bitbucket - just look how many clones of Exiv2 project exist nowadays over there).

#8 - 29 May 2016 12:59 - Taras Kushnir

Gilles Caulier wrote:

> digiKam use multithreading to do exactly the same task than you, and it do not crash.
> Gilles Caulier

It does not mean that it's stable, but means nobody found a race condition there.

If you will look in the code, multithreaded access to NsRegistry (properties.cpp) can lead to race condition: a reader who cycles over the std::map in lookupNsRegistry() and writer who calls registerNs() which modifies the very same std::map at the same time.

#9 - 29 May 2016 15:54 - Taras Kushnir

All the changes in patch summarized (+ fixes for Windows build)
https://bitbucket.org/ribtoks/xpiks-deps/commits/18181953b628ed56719fd8a8c1774eeac3a0999b?at=master

#10 - 29 May 2016 17:08 - Robin Mills

- File DSC_0042.jpg added

- Status changed from New to Assigned

- % Done changed from 0 to 50

- Estimated time set to 2.00 h

Thanks, Taras. I see you've been busy while I've been having a nice day out. I attach some flowers from Buxton, Derbyshire, England.

I have found it impossible to decrypt your urls (at least one of which is wrong) back into a patch that compiles. Can I ask you to make your changes to a checked out copy of the code and send me a copy of the output from the command:

```
$ svn diff
```

I see you have made changes to the CMake build to prefer C++11. This may be something you'd like to see us support, however it's unrelated to this issue. Can I ask you to submit a feature request to support C++11 and provide the patch. Supporting C++11 does not only involve CMake, it has implications for the msvc and ./configure build environments. I don't know if we should say "use C++11 if available". Is this the normal default or should it be build a build option (for both CMake and ./configure).

Are the changes to preview.cpp and epsimage.cpp related to C++11?

I will not get into a discussion of the superiority of GIT over SVN.

You're right. The foo_ statics in XMPsdk are not thread safe. I believe they came from the Adobe SDK. One of the priority features for v0.27 is to update our code to use XMPsdk as an external library and of course to adopt the latest Adobe XMPsdk.

DSC_0042.jpg

#11 - 29 May 2016 17:28 - Taras Kushnir

- File rwlock_025.svn_patch added

output from `svn diff` is in the attachment

using Exiv2::byte; (in preview.cpp and epsimage.cpp) is a compilation fix for ambiguity when building in Windows (some types header in windows has typedef for "byte");

Changes for C++11 in CMakeLists.txt are only related to my project and you can dismiss them (of course being able to build Exiv2 for C++11 without errors is vital). They are not included in the file attached to this message.

I need you also to review my changes - I'm not sure I didn't break something and also whether I covered all threading issues in the domain I patched (I mean it's possible that my changes can cause a deadlock - if one of the routines I refactored is somehow called from another - somebody should review it thoroughly)

#12 - 29 May 2016 18:57 - Robin Mills

I've added issue #1188 to support C++11.

You code is still failing to compile (as I found after I'd tried to build the patch manually). I've added the missing file rwlock.h, however there's a static rowlock_ missing from somewhere.

Still doesn't compile:

```
libtool: compile:  g++ -ggdb -fvisibility=hidden -fvisibility-inlines-hidden -Wall -Wcast-align \
-Wpointer-arith -Wformat-security -Wmissing-format-attribute -Woverloaded-virtual -W -MMD \
-I../src -I../include/ -I../include/exiv2 -I/usr/local/include -I/usr/local/include \
-DEXV_LOCALEDIR=\"/usr/local/share/locale\" -I../xmpsdk/include -c -DEXV_BUILDING_LIB=1 \
properties.cpp  -fno-common -DPIC -o .libs/properties.o
properties.cpp:2242:46: error: no member named 'rwLock_' in 'Exiv2::XmpProperties'
    Internal::RWLock          XmpProperties::rwLock_;
                              ~~~~~~~~~~~~~~~^
1 error generated.
make[1]: *** [properties.o] Error 1
make: *** [all] Error 2
525 rmills@rmillsmbp:~/gnu/exiv2/bigtiff $
```

I looked at your new file ***rwlock.hpp***. I think it's OK. We don't have anything in our test suite to test multi-threading, so I can only give it visual inspection. I will review it carefully before I submit it and after it and passes the test suite on all our supported platforms.

#13 - 29 May 2016 19:02 - Taras Kushnir

Robin Mills wrote:

> I've added issue #1188 to support C++11.
>
> You code is still failing to compile (as I found after I'd tried to build the patch manually). I've added the missing file rwlock.h, however there's a static rowlock_ missing from somewhere.
>
> Still doesn't compile:[...]I looked at your new file ***rwlock.hpp***. I think it's OK. We don't have anything in our test suite to test multi-threading, so I can only give it visual inspection. I will review it carefully before I submit it and after it and passes the test suite on all our supported platforms.

Compilation error is because of you didn't apply patch to properties.hpp where the rwLock_ is defined.
Pls take a look at the included file in the previous commit (rwlock_025.svn_patch)

Of course file rwlock.hpp is ok - it's super trivial and it's just wrapper over the library calls. I was talking about refactoring of properties.cpp. It has a lot of tricky methods which call one another - ns(), nsInfo() and friends.

#14 - 29 May 2016 19:33 - Robin Mills

- Status changed from Assigned to Closed

- % Done changed from 50 to 100

- Estimated time changed from 2.00 h to 3.00 h

I did apply the patch and failed to notice a message from properties.hpp "Failed to apply hunk". I did try a couple of times to add the static. Other compilation issues.

I will have to check out v0.25 and apply your patch to that to discover exactly what you have done. Then I can treat the trunk appropriately.

I'm closing this bug because you have a work-around (your patch). I am on vacation to get away from stuff like this. I've linked this issue to #941 to update the Adobe SDK.

#15 - 30 May 2016 07:33 - Taras Kushnir

Patch file was generated from differences from 0.25 release, but not from trunk. If this file was modified in trunk, of course there would be merge conflicts. I didn't generate it against trunk because I won't know how to correctly resolve merge conflicts (but you will).

I hope so much that this fix will get into 0.26

#16 - 30 May 2016 09:55 - Robin Mills

A fix for this will go into v0.26. I was too tired last night to deal with this. I've tried your patch on v0.25 and it builds and passes the test suite. I will add it to the trunk and it will ship in v0.26.

I'll review your code. I didn't write Exiv2, so there are parts that I know well, and other dark corners. Everything concerning XMP is in one of the dark places.

I'm not involved in the release process - that is exclusively the domain of the project founder Andreas. He is clever, cautious and careful. I'll be surprised if he has tests beyond the test suite, however I've never discussed it with him.

#17 - 30 May 2016 14:46 - Robin Mills

Fix submitted r4308. Thank You to Tara for the patch.

#18 - 30 May 2016 15:07 - Robin Mills

- Estimated time changed from 3.00 h to 4.00 h

I looked at your changes. I see you are "grabbing the lock" accessing/modifying the XMPsdk data and "releasing the lock". Seems OK to me. I don't think it's perfect. This strategy could lead to starving one or more threads. However it's better than crashing!

Perhaps you could review r4308 to see if I have carelessly lost any of your code when I added it to the trunk.

I should add a multi-threaded test application to our test suite. There have been several multi-threading issues reported in the last couple of years. If you have suitable code, I'll be happy to accept your help. There's lots to be done and I only have one (old) pair of hands.

Anyway - thank you for investigating and fixing this matter. Your help is appreciated. I have opened issue #1188 to provide support for C++11.

#19 - 30 May 2016 18:23 - Taras Kushnir

You have moved my changes well, but you didn't add rwlock.hpp to visual studio projects thus exiv2 will not compile under Windows. I didn't include it to the patch, because it was impossible - I've converted all projects to VS 2015.

Can you please elaborate on what starving you were talking about? If there would be so many readers that writers will never have access? Can you provide an example so I can fix flaws in my patch?

Yes, your test suite should definitely include threading tests.

I would contribute to exiv2 if only I didn't have tons of work on my own project. If I will find any new issues, I will try to fix them in exiv2. Anyway, you can point me out these threading issues so I can investigate whether they will affect me.

#20 - 30 May 2016 19:01 - Robin Mills

- % Done changed from 100 to 80

- Estimated time changed from 4.00 h to 5.00 h

You're right about MSVC. I'm surprised. I thought CL would find include/exiv2/rwlock.hpp without assistance from Visual Studio. Grrrrr. Something else to do. However you are right and the build failure has been reproduced on the buildserver. Our "on commit" msvc build is Visual Studio 2005/64. Our nightly build is msvc 2005/8/10/12/13/15 (Win32 and x64). I'll fix msvc later in the week.
http://exiv2.dyndns.org:8080/job/Exiv2-trunk/label=msvc/2456/console

Do you know the "dining philosophers" puzzle? This is a classic multi-threading puzzle involving two resources. Each philosopher has to acquire the knife ***AND*** fork to eat. Of course ***deadlock*** must be avoided, however ***starvation*** must always be avoided. Even with a single resource in which only the fork is required, in the mad scramble to acquire the fork, one philosopher could be unsuccessful in ever acquiring the fork. In a single process that will terminate in finite time, this will not happen because eventually all philosophers will acquire the fork and eat. Starvation is impossible.

However, there is an outstanding project (part of our webready initiative) to provide an exiv2d daemon/server. If exiv2d is very busy, he could starve a client. Regrettably the single lock model cannot solve starvation. It is necessary to add a priority mechanism to ensure that the most hungry philosopher gets to eat first.

Caution: the single lock model is very vulnerable to deadlock the instant anybody adds an additional lock anywhere in the system. Multi-threaded systems are not easy.

I worked on embedded systems at Novariant in Fremont, CA. They developed and use QF (Quantum Framework) to avoid deadlock, race and starvation by design. You can find out more about this here: http://www.state-machine.com

For sure, I don't want to complicate exiv2 by adding QF. Your contribution of rwlock.hpp seems sufficient for the moment and I thank you for doing this work. It's fine to leave things as they are. Thank You for your contribution.

And thank you for considering contributing to Exiv2. It's a simple fact that busy people are busy. It's not easy to recruit people to invest the effort involved in open source as I am sure you already know.

#21 - 30 May 2016 19:12 - Taras Kushnir

Yes, I know "dining philosophers" problem. However, in my understanding starvation is when low-priority thread constantly fails to acquire a resource because of context switches between higher-priority threads. I would not call starvation any situation when some threads are waiting some time in order to acquire the resource (as in this case - readers and writers are supposed to be of equal priority), because such waiting is inevitable part of live with any synchronization.

I completely agree that multithreading is super error-prone and especially for a stranger to a project to refactor some parts of it to multithreaded support. That's why I was asking to review my changes very carefully.

Thanks for the reference to Quantum Framework. Will take a look into that!

#22 - 30 May 2016 19:29 - Taras Kushnir

There's also another strange thing related to this issue.

Static methods I've mentioned predictably crash in OS X when being used with 30-40 threads, but never crash on Windows 7 and Windows 10 even on different computers. I can't understand why they don't crash.

#23 - 30 May 2016 19:49 - Robin Mills

- Status changed from Closed to Assigned

- % Done changed from 80 to 70

- Estimated time changed from 5.00 h to 7.00 h

Well, if you're confident that the thread scheduling algorithm embodies priority, then all is well ***if*** all threads are equal priority. The hungriest philosopher will get to eat and avoid starvation. Multi-threading is a process matter. Thread safety is a library prerequisite to be used in a multi-threaded application. There used to be an article on the Wiki about thread safety. It has mysteriously disappeared. I think we agree that libexiv2 is "mostly" thread-safe. We currently do not have a multi-threading example. Perhaps I should write one using QF. That would be fun.

The top feature requested for v0.27 is to upgrade our XMP support to use the latest Adobe SDK as an external library. Maybe there will a thread-safe fix for the XMPsdk ***fooBar***_ statics. I believe the XMPsdk code embedded in Exiv2 was added by Brad before I joined the project 8 years ago. Andreas said something like "anything involving XMP is a total pain". I'm about to discover the truth of that ***after*** v0.26 is code-complete in June and ***after*** I have a break from Exiv2 in July and August (to deal with various family matters).

I don't want to spend any more time on this at the moment (after from getting MSVC to build). If it's working on 40 threads on MacOS-X, I'm OK with that. Gimme'a'break, buddy. I'm on vacation!

#24 - 30 May 2016 19:54 - Taras Kushnir

Sure. Excuse me for distracting you more than necessary. Have a nice time.

#25 - 31 May 2016 07:15 - Taras Kushnir

https://bitbucket.org/ribtoks/xpiks-deps/commits/23231c5ab8edafd4e873634dae742996ef068e1e

One more fix needed for Windows build to succeed (bool -> BOOLEAN)

#26 - 31 May 2016 08:00 - Robin Mills

- File LargsPanorama.jpg added

Taras

The MSVC build is failing due to the code in the WIN32_ branch of rwlock.hpp. It doesn't require changes in the msvc project files. There are a couple of things which aren't quite right. Here's the code:

```
namespace Exiv2 {
    namespace Internal {
#ifdef _WIN32
        class RWLock
        {
        public:
            RWLock()
            {
                InitializeSRWLock(&rwlock_);
            }

            ~RWLock()
            {
                // do not explicitly destroy
            }

            void wrlock()
            {
                AcquireSRWLockExclusive(&rwlock_);
            }

            bool trywrlock()
            {
                return TryAcquireSRWLockExclusive(&rwlock_);
            }

            void rdlock()
            {
                AcquireSRWLockShared(&rwlock_);
            }

            bool tryrdlock()
            {
                return TryAcquireSRWLockShared(&rwlock_);
            }

            void rdunlock()
            {
                ReleaseSRWLockShared(&rwlock_);
            }

            void wrunlock()
            {
                ReleaseSRWLockExclusive(&rwlock_);
            }

        private:
            SRWLOCK rwlock_;
        };
#else
```

The error is on SRWLOCK rwlock_; Macro/class SRWLOCK isn't defined. I changed it to ScopedReadLock and added a ***class ScopedReadLock;*** forward reference. Compiler grumbled in RWLock::RWLock() c'tor about not initialising rwlock_.

I know you've got access to Visual Studio 2015. Can you take a look and let me know the fix please? We're off to Scotland today. I'll leave you to have a look while I'm on the road to the land of haggis, bagpipes and kilts. Forecast for Largs (where I was born) is 21ºC and full sun.

LargsPanorama.jpg

#27 - 31 May 2016 08:09 - Taras Kushnir

Robin, I've just compiled patched exiv2 v0.25 successfully with Visual Studio 2015 and yesterday I did the same with Visual Studio 2013.

https://msdn.microsoft.com/en-us/library/br244843.aspx - SRWLock is a known class. I've included <windows.h> so it should have been resolved. From your description it's not quite clear what can be wrong. I need to investigate the particular case. Maybe there're some other errors which screw SRWLock's definition and VC compiler thinks it's unknown class.

P.S. Me posting here does not oblige you to respond while you're on vacation. I'm just posting important stuff so it won't get lost with time.

#28 - 31 May 2016 09:19 - Robin Mills

Regrettably, SRWLock isn't in VS 2005/8/10/12. Maybe it's possible to use a static CRITICAL_SECTION to achieve our aims.
http://stackoverflow.com/questions/3498798/replace-critical-section-with-srw-lock

CRITICAL_SECTION is a Win32 feature, so you should be able to develop/test it with VS2015, even if you decide to use CRITICAL_SECTION in older versions of Visual Studio and SRWLock in 2013/15. There's code in include/exiv2/exv_msvc.h to detect the version of CL in use.

I don't feel obliged to do this on vacation *or at any other time*! However I like to deal with support issues quickly to let me focus on developing v0.26. I still have several things to finish for v0.26 and I'm in danger of slipping due to support issues consuming my time. I'm going to defer the low priority features of v0.26 to make the schedule.   http://dev.exiv2.org/news/2

#29 - 31 May 2016 09:26 - Taras Kushnir

It's possible to use CriticalSection, but it will be less optimal than Read Write lock. I would prefer to use Read/Write lock whenever is possible for VS 13/15, and use Critical Section in other cases. I will send a patch for that a bit later today.

#30 - 31 May 2016 15:41 - T Modes

The current code produces over 100 warnings of type
warning C4800: "BOOLEAN": forcing value to bool 'true' or 'false' (performance warning)
in rwlock.hpp(33) and rwlock.hpp(43) with MSVC2015.

Also function TryAcquireSRWLockExclusive is only available in Windows 7 and later. So the code will not run on Windows XP and Vista.

#31 - 31 May 2016 16:04 - Taras Kushnir

Pls see comment #25 - I've added one more tiny patch. dev.exiv2.org/issues/1187#note-25

TryAcquireSRWLockExclusive indeed will not work before Windows 7 ~~(but who cares)~~. Anyway, that part should be rewritten in order to be compilable by legacy visual studios (like vs 2005).

#32 - 31 May 2016 16:34 - T Modes

And it is not consistent:
On Windows (with your patch) we have in class RWLock
BOOLEAN trywrlock()
and otherwise
int trywrlock()
The function should not have different return types depending on the system.

#33 - 31 May 2016 21:24 - Taras Kushnir

Completely agree about consistency. As I've stated, didn't have much time since my changes with bug in exiv2 were in production so I wanted to fix that asap and afterwards sent my patch here.

How can I have access to the repo to fix all those things and not sending you "patches via email"? Are there in SVN world any kind of pull requests?

#34 - 31 May 2016 21:52 - Robin Mills

Taras:

I can't grant you write access to the repos. The credentials to grant write access are retained by Andreas and he likes new contributors to provide patches while we get to know each other.

At the moment, MSVC 2005/8/10/12 builds are broken on the trunk by r4308. We should fix or revert this. I can setup a branch for rwlock development and when the branch builds and passes the test suite on all platforms, I'll merge to trunk.

I agree with T about consistency. BOOLEAN is a Windows 1.0 relic. It's a macro (or typedef) for *unsigned short int* or something. I would prefer that you return bool, even if it is Internal and not exposed as a public API.

#35 - 01 Jun 2016 08:13 - Taras Kushnir

https://bitbucket.org/ribtoks/xpiks-deps/commits/10ae89e8fca5e472a5779ad30398da74e9ccb0f7 - this is the patch with consolidation of api and workaround for Windows < 7

Why do you guys care to support Windows Vista and XP? They are not supported by MS anymore..
~~(from the other point, why do I even ask.. you still have VS 2003 and SVN..)~~

#36 - 01 Jun 2016 09:39 - Robin Mills

Thanks, Taras. I'll look at that this evening. Suggestion:

```
bool trywrlock()
            {
                return 0 != TryAcquireSRWLockExclusive(&rwlock_);
            }
```

Change to:

```
bool trywrlock()
            {
                return TryAcquireSRWLockExclusive(&rwlock_) == TRUE;
            }
```

GIT vs SVN

http://dev.exiv2.org/boards/3/topics/1786?r=1788#message-1788

If you join me in Exiv2, I will agree to use GIT for v0.27 ***IF you promise*** to help me when I get stuck in GIT hell.

Old Versions of Visual Studio:

If we can build on old versions of Visual Studio, I'm happy to continue to support them. When I was contracting in Silicon Valley, I was rather surprised to discover ancient old tools being used on legacy systems (Dev Studio 6.0 and the like). I don't want to encourage this, however if it's not painful to support 2005/8/10/12, let's keep them alive. v0.26 will be the final release that support Visual Studio .Net (2003).

#37 - 01 Jun 2016 10:19 - Taras Kushnir

I don't agree with your proposed change.
TRUE is defined as 1 in wintypes.h

```
#define FALSE 0
#define TRUE 1
```

while the MSDN about TryAcquireSRWLockShared says: https://msdn.microsoft.com/en-us/library/windows/desktop/dd405524%28v=vs.85%29.aspx

```
*Return value*
If the lock is successfully acquired, the return value is nonzero.
```

Obviosly, nonzero != 1 and I would leave it as is for the compatibility with the official documentation.

Such a discussion about method which is even not used. It will make more sense to remove it until it will be used.

Git stuff. Please define "help". I can offer my help in setting up and maintaining git repo at GitHub + continuous integration for Windows (AppVeyor) and Linux/OS X (Travis CI). However I'm not sure what do you mean by "help".

I also would like you to understand, that I'm not saying that "Git is super the best. Just use Git". What I'm saying instead is that nowadays there're a lot of usefully interconnected services which in most cases work over Git(Hub). Also accepting and managing contributions in there is so much easier .

(Besides that Git seems to have it's own pros and cons comparatively to SVN and seems to be more useful for me, but it's just my point of view)

#38 - 01 Jun 2016 10:30 - Robin Mills

What do I mean by "Help?". Folks get into trouble with GIT. I've worked with folks who use GIT and they say "It's months of mental fog before you 'get git'". Within a few commands you can be lost and have no idea of the state of your code. So they pull down a fresh copy from the repos. I'm a little afraid that I'll loose control of the code. There is no "central" server. So if I'm in trouble, I don't know from where to pull down a new copy.

SVN is very stable and I have nice tools (SmartSVN). I have purchased a license for SmartGIT. I want to concentrate on Exiv2 development and avoid battles with the source code control system.

My thoughts about moving to GIT are to get involved with a well established project that uses it. I'll come up to speed in the company of more experience GIT users. I'm nervous to drag Exiv2 down this road on my own.

#39 - 01 Jun 2016 10:33 - Robin Mills

I've misunderstood the return value of that function. You are right and I am wrong. Apologies.

#40 - 01 Jun 2016 11:56 - Taras Kushnir

Regarding what do you say about "lost in few commands" - my experience with Git is directly opposite. For me with Git it's always clear where you are.

In my opinion GUI "tools/helpers" for git are the things that will mislead you in the first place since the reason you're using them is that you don't want to bring changes in your life and have exactly UI as you've had with SVN and don't care that Git is underneath. As for me, I didn't ever use any third-party tool for git except of the tools which go with standard git distribution and don't feed need in one.

I've just been recently through move from SVN to Git in quite a big organization with people working with SVN half of their lifes. It was painful for them. But now, after 3 months, majority of them sometimes say "huh, this or that is so much easier now".

Anyway, I will be able to help you with porting exiv2 from svn to git when you will ask particular questions (better in private, of course, since discussion of this issue went too far). To be confident with Git I can suggest you to read only one book "Pro Git" and I hope it will be enough.

#41 - 01 Jun 2016 16:47 - Robin Mills

Yes. I bought an e-book license for "Pro Git". Thanks for being willing to help with Git. I have no intention to use Git as it will consume lots of my time that is better spent working on exiv2.

Thanks for your Windows Patch for r4308. I'll review that and submit it after dinner.

#42 - 07 Jun 2016 21:25 - Robin Mills

- % Done changed from 70 to 90

- Estimated time changed from 7.00 h to 15.00 h

Quite a lot of work necessary to stabilise the build on all platforms and the buildserver.

#43 - 08 Jun 2016 04:48 - Taras Kushnir

Thank You, Robin for putting in the effort to get my patch into libexiv2.

#44 - 08 Jun 2016 09:22 - Robin Mills

- Status changed from Assigned to Closed

- % Done changed from 90 to 100

Buildserver built and tested with CMake: http://rmillsmm:8080/job/trunk-cmake-daily/275/
Buildserver built and tested with Native tools (./configure and msvc2005 project/solution files): http://rmillsmm:8080/job/Exiv2-trunk/2462/

MinGW build has been broken since late March and I will investigate/remedy that shortly.

#45 - 24 Jul 2016 09:01 - Robin Mills

I have received the following email from Taras:

> Unfortunately, I've found a bug in my patch. 1 odd lock aquisition.
> https://bitbucket.org/ribtoks/xpiks-deps/commits/f4ddc6129174515dbe87043a9c9669982f1e2a62 - link to a 1-line patch for a patch.
>
> If that's not gonna work for you, the patch is to remove the line:
>
> Internal::ScopedReadLock srl(rwLock_);
>
> in file properties.cpp in function
>
> `const XmpNsInfo* XmpProperties::nsInfoUnsafe(const std::string& prefix)`

And my reply:

> Thank you for the patch which has been submitted as r4350.

Files

DSC_0042.jpg	110 KB	29 May 2016	Robin Mills
rwlock_025.svn_patch	6.25 KB	29 May 2016	Taras Kushnir
LargsPanorama.jpg	122 KB	31 May 2016	Robin Mills