Exiv2 - Bug #846

Porting the Video Code to MSVC (umbrella)

03 Sep 2012 00:30 - Robin Mills

Status:	Closed		**Start date:**	02 Sep 2012
Priority:	Normal		**Due date:**	
Assignee:	Robin Mills		**% Done:**	80%
Category:	build		**Estimated time:**	0.00 hour
Target version:	0.24			

Description

Abhinav

I'm porting your code to MSVC (32 and 64 bit). Let's use this issue to track our progress (that's what "umbrella" means)

Good News:
I've got everything to compile and link (32/64 static/DLL Release/Debug) You're code's very clear and well formatted. Good Job!

Ungood News:
I've had to throw a lot of casts for (size_t) and (long) into your code. I'm rather bothered by this. I think the main issue is that the I/O APIs use long and maybe they should be size_t or possibly uint64_t. For example: io->read(), io->seek(). For images, I think it would be rather unusual to have a HUGE (>2Gbyte) file, for video this could be common. However changing the I/O APIs will require lots of changes elsewhere in the code AND will impact user applications. Ummmm. I really ought to ask Andreas for this opinion about this.

From here:
I'll probably post a patch here tomorrow to show you what I've done. It's mid-night here and I'm off to bed. Tomorrow (Labor Day Holiday), I'll run the test suite on Windows and try the patch on Mac/Linux to see what happens.

Anyway, I'm wondering how much time you have to work with me on this? Are you available - or busy with something else?

Associated revisions

Revision 2934 - 04 Nov 2012 01:02 - Robin Mills

Fix: #846. Fix for video-test errors (about aspectRatio) on MSVC 2003. See #846 message 20 for discussion.

History

#1 - 03 Sep 2012 12:04 - Abhinav Badola

Hi Robin,

   Anyway, I'm wondering how much time you have to work with me on this? Are you available - or busy with something else?

I will try my best to keep up with all the coding that is required in the concerned bug. Honestly, I have no knowledge about porting the code to MSVC environment, but you may guide me through the process, I assure you that I will catch up and learn quickly.
Though I wont be able to be available all the time, due to college classes and assignments, but I will try my best to contribute to the code as and when I get time. I am generally free on weekends and I can work on the code during weekends.

#2 - 03 Sep 2012 13:47 - Robin Mills

Thanks for getting back to me. You don't need to know anything about MSVC. However I hope you'll be able to review changes I make to your code. I don't expect to need much of your time.

I've been looking at this again today. I can't get the code base to build on Linux/Cygwin/Mac. I ran my MSVC build (from last night) on the test suite and got hundreds of errors.

I think I'll do nothing about MSVC for the moment. It only took 2-3 hours last night to update your code to build and link (both 32 and 64 bit). I'll put the porting project on hold until the linux (and mac and cygwin) builds are stable and passing the test suite. When we're rock stable, I'll discuss the 64 bit I/O API with Andreas. If that's added, I suspect your code will only require minor changes.

So... I'm glad you are available. That's good to know. Rest assured, I don't want much of your time (probably only a few hours). I think it might be a little time (a month or two) before I return to this matter.

#3 - 03 Sep 2012 16:36 - Robin Mills

- Assignee changed from Abhinav Badola to Robin Mills

#4 - 03 Sep 2012 19:03 - Andreas Huggel

Abhinav, "porting" here simply means (a) to make the code compile on MSVC and (b) to update the MSVC build scripts. For (a) I expected pretty much exactly what Robin said in his report. The MSVC compiler is more verbose about type conversions than g++ (it issues warnings) and since we have warnings set to errors in the MSVC build environment we need to fix them all. That's a good thing. I don't think there is a need for any Windows specific code, we already have that covered where necessary for this (e.g., IO, charset conversion).

Robin, I'm fine if you just add casts for now, just please use C++ casts, so we can find them more easily. I've discussed the type correctness issue with Abhinav before, we're aware that this needs work and have started looking at this. Your comment about the IO functions using long instead of size_t also came up I think. It's like this for historic reasons (even some of the underlying C functions use long), I prefer to stick with what we have rather than change the IO interface (that may trigger a tsunami even within the library). That means there will be some casts left eventually (just like elsewhere in the code), as long as we can minimize them and know why we need each of them I'm ok.

Andreas

#5 - 03 Sep 2012 21:58 - Robin Mills

Andreas:
I share your concern about the 64 bit I/O, so let's leave it at 32bit. I'll do the port again (in a few weeks) with C++ casts and I expect that to be quite straight-forward.

#6 - 05 Sep 2012 10:52 - Abhinav Badola

All right..!!
I am in for the job whenever I am needed.

#7 - 14 Sep 2012 19:58 - Robin Mills

I've updated the project files in the msvc and msvc64 directories SVN:2875.

msvc:
builds and passes the sanity test (runner.bat).
doesn't do well on the test suite (cygwin/unix/mac aren't good at present either)

msvc64:
32 bit builds are OK (and pass sanity)
64 bit builds do not compile (so cannot be tested)

The problems with msvc64/64 bit are related to 32/64 bit APIs.

I have been able to get the code to compile I using many casts (100s of them). Andreas has requested that we use C++ casts as they are easier to identify in the code (for review and future elimination). I've had another couple of thoughts about this:
1) Introduce 64 bit I/O API's for MSVC only in the header files.
- use #if _MSC_VER ... #endif to reveal them to MSVC
- put inline functions in the header files to do the downcasting
- don't change anything in the C++ files

2) Use a MACRO to define a cast (or casts). The exact contents of the cast can be changed without disturbing the code.

Approach (1) is elegant if can be made to work. It means that MSVC related changes are local and easily identified. Changes to the CPP code can be made without having to think "what about MSVC".

Shawn has offered to work on this task. Shawn - welcome to the team.

#8 - 18 Sep 2012 00:18 - Robin Mills

Shawn sent me a very nice patch file with changes. All changes were C++ casts. A very nice job. Thank you, Shawn.

I've reviewed and simplified Shawn's changes by defining MSVC specific 32 bit functions in basicio.hpp (as discussed in #7 above). This reduced the number of changed files. It compiles, so I've submitted this SVN:2879. I'll test it in the next couple of days.

```
C:\gnu.video\exiv2\src>svn status q | sort
M       asfvideo.cpp       <-- Abhinav please review
M       basicio.cpp        <-- Shawn's changes
M       basicio.hpp        <-- Robin added new seek functions
M       jp2image.cpp       <-- Shawn added a cast
M       matroskavideo.cpp  <-- Abhinav please review
M       quicktimevideo.cpp <-- Abhinav please review
M       riffvideo.cpp      <-- Abhinav please review
```

Abhinav:
Can you review the changes in your files, please? There are some changes to offsets in arrays that surprise me.

Robin

#9 - 18 Sep 2012 01:36 - Shawn Jean

- File shawn_test_result.txt added

You guys did a awesome project. Well designed, easy to understand! Quite good to join you.

After my changies to it, somethings goes wrong with the matroskavideo.cpp. I didn't figure it out. Abhinav could you please review it.

#10 - 18 Sep 2012 11:51 - Abhinav Badola

Hi Shawn, Robin,

Right now I am a bit busy with some college assignments.
I will definitely review the code by this Thursday.

#11 - 19 Sep 2012 14:01 - Abhinav Badola

Hi Robin,
I was going through the code, and I guess some new casts and function overloading are creating some problem.

The problem is majorly in basicio.cpp file.
I guess it would be compiling on Windows OS, but it is not compiling under Linux OS.

Here is the error log for further debugging-

/home/badola/trunk/src/bmpimage.cpp: In function 'bool Exiv2::isBmpType(Exiv2::BasicIo&, bool)':
/home/badola/trunk/src/bmpimage.cpp:161:40: error: call of overloaded 'seek(int, Exiv2::BasicIo::Position)' is ambiguous
/home/badola/trunk/src/bmpimage.cpp:161:40: note: candidates are:
/home/badola/trunk/src/basicio.hpp:172:21: note: virtual int Exiv2::BasicIo::seek(long int, Exiv2::BasicIo::Position)
/home/badola/trunk/src/basicio.hpp:181:21: note: virtual int Exiv2::BasicIo::seek(uint64_t, Exiv2::BasicIo::Position)
/home/badola/trunk/src/basicio.cpp:262:10: warning: unused parameter 'src' [-Wunused-parameter]
make[2]: * **[src/CMakeFiles/exiv2lib.dir/bmpimage.cpp.o] Error 1**
make[2]: Waiting for unfinished jobs....
/home/badola/trunk/src/basicio.cpp: In member function 'virtual int Exiv2::FileIo::seek(uint64_t, Exiv2::BasicIo::Position)':
/home/badola/trunk/src/basicio.cpp:851:10: error: 'fseeko' is not a member of 'std'
/home/badola/trunk/src/basicio.cpp:851:10: note: suggested alternative:
/usr/include/stdio.h:766:12: note: 'fseeko'
/home/badola/trunk/src/basicio.cpp: In member function 'virtual int Exiv2::MemIo::seek(uint64_t, Exiv2::BasicIo::Position)':
/home/badola/trunk/src/basicio.cpp:1154:16: warning: comparison of unsigned expression < 0 is always false [-Wtype-limits]
/home/badola/trunk/src/basicio.cpp:1154:34: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
/home/badola/trunk/src/basicio.cpp: In member function 'virtual int Exiv2::FileIo::seek(uint64_t, Exiv2::BasicIo::Position)':
/home/badola/trunk/src/basicio.cpp:853:2: warning: control reaches end of non-void function [-Wreturn-type]
make[2]: **[src/CMakeFiles/exiv2lib.dir/basicio.cpp.o] Error 1**
/home/badola/trunk/src/asfvideo.cpp: In member function 'virtual void Exiv2::AsfVideo::readMetadata()':
/home/badola/trunk/src/asfvideo.cpp:323:34: error: call of overloaded 'seek(int, Exiv2::BasicIo::Position)' is ambiguous
/home/badola/trunk/src/asfvideo.cpp:323:34: note: candidates are:
/home/badola/trunk/src/basicio.hpp:172:21: note: virtual int Exiv2::BasicIo::seek(long int, Exiv2::BasicIo::Position)
/home/badola/trunk/src/basicio.hpp:181:21: note: virtual int Exiv2::BasicIo::seek(uint64_t, Exiv2::BasicIo::Position)
/home/badola/trunk/src/asfvideo.cpp: In function 'bool Exiv2::isAsfType(Exiv2::BasicIo&, bool)':
/home/badola/trunk/src/asfvideo.cpp:742:37: error: call of overloaded 'seek(int, Exiv2::BasicIo::Position)' is ambiguous
/home/badola/trunk/src/asfvideo.cpp:742:37: note: candidates are:
/home/badola/trunk/src/basicio.hpp:172:21: note: virtual int Exiv2::BasicIo::seek(long int, Exiv2::BasicIo::Position)
/home/badola/trunk/src/basicio.hpp:181:21: note: virtual int Exiv2::BasicIo::seek(uint64_t, Exiv2::BasicIo::Position)
make[2]: [src/CMakeFiles/exiv2lib.dir/asfvideo.cpp.o] Error 1
make[1]: **[src/CMakeFiles/exiv2lib.dir/all] Error 2**
make: * [all] Error 2

#12 - 19 Sep 2012 21:28 - Robin Mills

Abhinav

Thanks for looking at this so promptly. You discovered I've broken the Unix before me! I've submitted SVN: 2880 to fix this. The code builds OK for me on GCC/cygwin/Linux (and with DevStudio 2005 using msvc and msvc64). It passes the test suite with much the same results that Shawn reported.

I'm having no success building with the automake files (make config ; ./configure ; ....). I can't get this to work on Mac/Linux/Cygwin. It complains about can't find exv_conf.h reports 1000's of type issues.

However it builds OK with cmake (on Linux and Cygwin)
$ cd exiv2
$ mkdir build
$ cd build
$ cmake ..

$ make

Are you or Shawn having any difficulty with the automake build? This has been rock stable for years (although I made a change for cygwin recently). I'll investigate at the weekend.

#13 - 20 Sep 2012 22:27 - Robin Mills

I've raised an issue [#849](#) about the issue with ./configure. I fixed it SVN: 2881 & 2882.

As far as I can see, the code's now building and running the test suite on:

- ./configure on Linux/Mac/Cygwin
- MSVC/2005 in both msvc and msvc64
- CMake/makefiles Linux/Mac/Cygwin

I haven't looked at CMake/MSVC since June. CMake support is still in development - however it our goal to adopt CMake as our only build environment.

Abhinav:
I'm sure you have plenty of course work. If you're interested in a little software challenge, write a script to "calculate" the dependencies of exv_conf.h. It wouldn't be difficult, and probably fun. Use grep to find the includes in every file in the src directory and use tsort to put them into order. The compiler dependency code does this (and generates the mysterious .d files). The compiler is a much stronger parser than grep of course. However the tsort phase will be about the same. Show it to your professor. I'm sure he'll be impressed.

Robin

#14 - 23 Sep 2012 12:13 - Abhinav Badola

Hi Robin,

> Abhinav:
> I'm sure you have plenty of course work. If you're interested in a little software challenge, write a script to "calculate" the dependencies of exv_conf.h. It wouldn't be difficult, and probably fun. Use grep to find the includes in every file in the src directory and use tsort to put them into order. The compiler dependency code does this (and generates the mysterious .d files). The compiler is a much stronger parser than grep of course. However the tsort phase will be about the same. Show it to your professor. I'm sure he'll be impressed.

As for this script, I think Andreas fix on bug [#849](#) would be sufficient.
Is there still a need to develop the script..?

#15 - 23 Sep 2012 18:30 - Andreas Huggel

Abhinav Badola wrote:

> Hi Robin,
>
> > Abhinav:
> > I'm sure you have plenty of course work. If you're interested in a little software challenge, write a script to "calculate" the dependencies of exv_conf.h. It wouldn't be difficult, and probably fun. Use grep to find the includes in every file in the src directory and use tsort to put them into order. The compiler dependency code does this (and generates the mysterious .d files). The compiler is a much stronger parser than grep of course. However the tsort phase will be about the same. Show it to your professor. I'm sure he'll be impressed.
>
> As for this script, I think Andreas fix on bug [#849](#) would be sufficient.
> Is there still a need to develop the script..?

I only saw this yesterday after I made changes.. What I committed is a pragmatic approach, it should solve the maintenance issue and is as simple as it gets. That is good for the project, but your professor wouldn't be impressed :)

#16 - 23 Sep 2012 19:12 - Robin Mills

There is no need for the script. Everything is fine with my fix, and Andreas' even better fix. Dependency analysis is an interesting matter and I was suggesting a little project you might find fun. However, there is not need for you to bother with this - it was simply a challenge to sharpen your brain!

The test harness is producing messages concerning the video code and I'd like to ask you to investigate those. I was surprised that Shawn made changes to array offsets in your code with his submission and perhaps those changes introduced the messages. Fixing this is important, but not urgent.

#17 - 19 Oct 2012 21:41 - Robin Mills

- % Done changed from 0 to 100

I'm going to mark this as 100% resolved as I believe that MSVC is building the video code in a satisfactory manner. There is an outstanding matter that the video code is not passing the test suite, however I believe this occurs on all platforms and not only MSVC. I'll open a different issue about the

video test code an assign it to Abhinav.

#18 - 19 Oct 2012 22:03 - Robin Mills

- Status changed from Assigned to Resolved

I believe this is complete. I've changed the status to resolved. This issue won't be closed until the 0.24 release (expect late November 2012)

#19 - 03 Nov 2012 10:58 - Robin Mills

- Status changed from Resolved to Assigned

- % Done changed from 100 to 80

Thanks to Abhinav, the video test failures have been investigated and resolved. (Issue: #862).

Issues have appeared with DevStudio 2003 builds failing the test suite. Specifically the aspectRatio in a couple of the test files is being incorrectly calculated and reported. MSVC/2003 Release and ReleaseDLL are impacted. All builds with msvc64 builds pass on DevStudio 2005/2008/2010. I'll investigate this issue as I know that Abhinav does not have the tools to work on this.

#20 - 04 Nov 2012 01:16 - Robin Mills

I've fixed this SVN:2934

The aspectRatio() member functions in the 4 video classes used floating point comparison to determine xmpData_["Xmp.video.AspectRatio"]. I've replaced this with an integer switch and everything seems OK on msvc and msvc64 build environments. I'll test it tomorrow on Mac and Linux.

The code contains the comment (presumably from Abhinav)

```
    // TODO – Make a better unified method to handle all cases of Aspect Ratio
```

We could:

1. Push this code into the Image base class to avoid code duplication.
2. Use a map<integer,string> to avoid the switch completely.

I'll don't think I will do either. Changing the Image base class involves risk to all Image classes. If this passes on Mac/Linux, I'm going to consider the video/aspectRatio/DevStudio issue fixed.

#21 - 23 Jan 2013 21:35 - Robin Mills

- Status changed from Assigned to Resolved

#22 - 24 Jul 2013 15:25 - Robin Mills

- Status changed from Resolved to Closed

Fixed in 0.24.

Files

shawn_test_result.txt	8.31 KB	18 Sep 2012	Shawn Jean