

## Exiv2 - Feature #1121

### Visual Studio support for v0.27

24 Sep 2015 07:32 - Robin Mills

<b>Status:</b>	Closed	<b>Start date:</b>	24 Sep 2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Robin Mills	<b>% Done:</b>	100%
<b>Category:</b>	build	<b>Estimated time:</b>	10.00 hours
<b>Target version:</b>	0.27		
<b>Description</b>			
Complete support for VS 2013 and 2015 including the build server.			
In v0.24, the build infrastructure from msvc2005 broke on Visual Studio 2012. I only discovered this during final testing in July 2013 and I quickly resolved this by introducing a new msvc2012 tree. However, I re-engineered msvc2005 for v0.25 to work correctly with Visual Studio 2005/8/10/12. There was insufficient time on v0.25 to investigate Visual Studio 2013, although several users reported success.			
I hope to continue to support msvc2003 for use by Visual Studio 2003 which does not have a 64 bit compiler. However msvc2003 will not support features such as webready which require curl and other compiler support. I hope to rename msvc2005 as msvc and it will support 2005/8/10/12/13/15 for 32 and 64 bit builds.			
v0.25 introduced a new script 'configure.bat' which calls configure.py. This enables the user to configure the Visual Studio build system to optionally select different libraries to be included in the build. Currently, when a library is selected, the code must be present in the prescribed directory relative to <exiv2dir>. This should be extended to enable the use of pre-build binary libraries. This will have a hugely beneficial impact on build times. When building support libraries from source, the increase in build time for expat and zlib are about 20-30 seconds each. However, openssl/libssh/curl adds about 20-30 minutes to the build. Storing precompiled copies of built libraries on build server will have a huge performance gain on the build server.			
The use of the clang compiler from within Visual Studio (as an alternative to the Microsoft cl compiler) should be investigated and supported if possible.			
The build server runs the test suite from jenkins_build.sh. I will investigate modifying the Visual Studio solution/project files to enable tests to be run from Visual Studio itself. To actually use those tests will require cygwin to be installed on the user's computer. When cygwin is not available, the user will be instructed to install cygwin. I had a prototype of this running some time ago and this is quite easy.			
When we add the groups of tests, the number of sub-projects in Visual Studio will be more than 50. A hierarchy of projects should be established. These would include "support" (zlib, expat etc) "core" (xmpsdk, src/*.cpp, exiv2, exifprint etc), "test apps" (iotest etc), "sample apps" (exiv2json etc), and "testing" (tests, teste etc).			
<b>Related issues:</b>			
Related to Exiv2 - Feature #1041: CMake toolchain for windows		<b>Closed</b>	<b>15 Mar 2015</b>
Related to Exiv2 - Support #1169: Validate flag EXV_UNICODE_PATH/msvc2005 bui...		<b>Closed</b>	<b>18 Mar 2016</b>
Related to Exiv2 - Bug #1171: localtime.c causes problems in MSVC 2015 & is n...		<b>Closed</b>	<b>22 Mar 2016</b>
Related to Exiv2 - Bug #1174: Visual Studio Support Enhancements		<b>Closed</b>	<b>31 Mar 2016</b>

#### Associated revisions

##### Revision 4065 - 05 Dec 2015 17:25 - Robin Mills

#1121 (msvc support for v0.26). exiv2 -vVg ^version reports version=8.00 (2005/x64) or version=8.00 (2005/x86) instead of version=8.00 (2005)

##### Revision 4685 - 21 Nov 2016 10:54 - Robin Mills

#1121 Fixes to msvc/\*.vcproj to support VS2017 RC!

#### History

##### #1 - 24 Sep 2015 07:37 - Robin Mills

- % Done changed from 0 to 10

- Estimated time set to 100.00 h

I've made a solid start on this. Visual Studio 2013 and 2015 are building and passing the test suite. [r3967r3966r3965r3960](https://github.com/exiv2/exiv2/commit/r3967r3966r3965r3960)

## #2 - 05 Dec 2015 17:35 - Robin Mills

- % Done changed from 10 to 20

I've made good progress with [#1041](#) (CMake Support). Basic exiv2+expat+zlib are building and passing the test suite. curl/openssl/libssh also build (well, openssl is prebuilt). I find it useful to monitor the progress of the build with `grep ^version build/rebuild.txt` which shows the version, but not x64/x86. I've submitted a change [r4065](#).

I've also investigated changing `configure.py` to accept the arguments `--with-curl[=directory]`. This is possible with the `argparse` module as follows:

```
import sys
import argparse

def main():
    parser = argparse.ArgumentParser(description='Example with long option names')

    parser.add_argument('--bool', action="store_true", default=False)
    parser.add_argument('--arg' , action="store", dest="arg")
    parser.add_argument('--ints', action="store", dest="ints")
    parser.add_argument('--foo' , nargs='?', const='default', default='')

    print(parser.parse_args( sys.argv[1:] ))

if __name__ == "__main__":
    main()
```

## #3 - 31 Mar 2016 13:13 - Robin Mills

- % Done changed from 20 to 50

- Estimated time changed from 100.00 h to 50.00 h

I'm going to defer most of this into v1.0 and some of it will be done in v0.27. The effort on Visual Studio/CMake and CMake/Visual Studio/Build server has been very considerable. So I'm drastically reducing the scope of this issue and it will recommence under [#1174](#) for v0.27.

## #4 - 13 Jun 2016 06:23 - Robin Mills

- Target version changed from 0.26 to 0.28

Deferred to v0.27 to make the schedule for v0.26.

## #5 - 13 Jun 2016 06:51 - Robin Mills

- Subject changed from Visual Studio support for v0.26 to Visual Studio support for v0.27

This was originally intended for v0.26. Has been deferred to make the schedule.

## #6 - 21 Nov 2016 10:55 - Robin Mills

I've built Exiv2 v0.26 RC1 with Visual Studio 2017 RC!

I've discovered some issues in our project files concerning paths for some header files [r4685](#)

VS2017 RC! doesn't work very well. The project convertor is crashes. If you reload the project several times and say "Don't override files", it will load and build. It suggests to enabling something called "Lightweight Load". Once selected, build clean/rebuild etc is totally broken. Let's hope the shipping version of VS2017 is better.

VS2017 will ship a new feature called **vcpkg**. This looks very promising in the demo and enables libraries to be downloaded, built and installed from a simple command-line such as:

```
$ vcpkg install exiv2-x86-static
```

Additionally, you can link those libraries simply by referencing the library include file from C++. I believe we have to submit a Vcpkg configuration to Microsoft. I suspect this will build exiv2 with zlib and expat. How well vcpkg can cope with options for webready (to build and link libssh, curl and openssl) remains to be discovered in v0.27.

VS2017 will have builtin support for CMake. I'm not optimistic in view of very painful experience with CMake/Visual Studio. Daniel and Gilles (who love and demanded Exiv2/CMake support) are welcome to investigate this for v0.27. I will not waste another second of my life on CMake/Visual Studio.

## #7 - 14 Nov 2018 16:44 - Robin Mills

- *Status changed from Assigned to Closed*
- *Target version changed from 0.28 to 0.27*
- *% Done changed from 50 to 100*
- *Estimated time changed from 50.00 h to 10.00 h*

This matter is closed as we have adopted CMake (and conan) to build Visual Studio. This is documented in README.md and README-CONAN.md