Exiv2 - Feature #1188

Provide build support for C++11

29 May 2016 17:39 - Robin Mills

Status:	Closed	**Start date:**	29 May 2016
Priority:	Normal	**Due date:**	
Assignee:	Robin Mills	**% Done:**	100%
Category:	build	**Estimated time:**	1.00 hour
Target version:	0.27		

Description

Related issues:

Related to Exiv2 - Bug #1187: Crash while reading in parallel threads	**Closed**	**29 May 2016**
Related to Exiv2 - Bug #1270: Using libexiv2.a/.lib in multhreaded app segfa...	**Closed**	**13 Jan 2017**

Associated revisions

Revision 4531 - 21 Sep 2016 18:46 - Robin Mills

#1188 set -std=c++98 for v0.26. I'll work on C++11 in v0.27. C++11 Deprecates AutoPtr, so supporting C++11 requires some effort.

#1188 Discusses CMake/C++11 in some detail. For v0.26, CMake also sets -std=c++98 r4530

Revision 4534 - 21 Sep 2016 22:21 - Robin Mills

#1188 and #1109 Correction to r4530 and r4531 to fix cygwin build-breaker (gnu++98 required for snprintf support)

History

#1 - 30 May 2016 09:40 - Robin Mills

In the discussion in #1187, Taras proposed the following change to CMakeLists.txt:

```
...
OPTION( EXIV2_ENABLE_CURL          "USE Libcurl for HttpIo"                              ON  )
OPTION( EXIV2_ENABLE_SSH           "USE Libssh for SshIo"                                ON  )

+include(CheckCXXCompilerFlag)
+CHECK_CXX_COMPILER_FLAG("-std=c++11" COMPILER_SUPPORTS_CXX11)
+CHECK_CXX_COMPILER_FLAG("-std=c++0x" COMPILER_SUPPORTS_CXX0X)
+if(COMPILER_SUPPORTS_CXX11)
+  set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -std=c++11 -stdlib=libc++")
+  message(STATUS "Using C++11")
+elseif(COMPILER_SUPPORTS_CXX0X)
+  set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -std=c++0x -stdlib=libc++")
+  message(STATUS "Using C++0x")
+else()
+  message(STATUS "The compiler ${CMAKE_CXX_COMPILER} has no C++11 support. Please use a different C++ compile
r.")
+endif()
+
+
IF( MINGW OR UNIX ) ...
```

This patch says "if the compiler supports C++11, use it. I'm not sure that this is desirable. This patch doesn't enable the user to say "don't use C++11 even if it is supported". We need to consider build and test for:

1. C++11 support for GCC/Clang and various editions of CL (msvc)
2. add an option for C++11 for ./configure, cmake and configure.py (msvc)
3. build with C++11 occasionally on the buildserver (once a week or so)

This seems like very good candidate feature for v0.27. http://dev.exiv2.org/news/3

#2 - 30 May 2016 17:53 - T Modes

CMake 3.1 and later has already some support for C++11. Instead of testing on your own (and missing of msvc) have a look at CMAKE_CXX_STANDARD in CMake doc.

So
SET(CMAKE_CXX_STANDARD 11)
in CMakeLists.txt activates the support for C++11 if available by the compiler. Otherwise it will fall back to the lower C++ standard.
If you want to control the behavior wrap it into an if statement.
(Only drawback you will need to update the
CMAKE_MINIMUM_REQUIRED(VERSION 3.1)
But this should not be a problem as CMake is currently at 3.5.2.)

#3 - 30 May 2016 18:10 - Robin Mills

Thanks. As you know, I am not a fan of CMake and will not be working on this issue until v0.26 ships. If you'd like to contribute to this, I will be delighted to accept your help. I have no issue with setting the minimum CMake version to 3.1.

#4 - 31 May 2016 06:36 - Taras Kushnir

At least under OS X 10.10.5 it's not enough just to add

```
SET(CMAKE_CXX_STANDARD 11)
```

since "-stdlib=libc++" parameter will not be added to CXX_FLAGS. That was the reason I've added bunch of duplicating stuff.

#5 - 31 May 2016 16:03 - T Modes

- File cxx11.diff added

Attached patch added a new option to CMake: EXIV2_ENABLE_CXX11.
By default it is off. When set to ON it will compile with C++11 mode of the compiler.

Concerning the OS X issue. Try first if the issue is fixed in current CMake (you did not specify your version).

#6 - 31 May 2016 16:07 - Taras Kushnir

My CMAKE version is quite recent if not the most recent.

```
> /Applications/CMake.app/Contents/bin/cmake --version
cmake version 3.5.2

CMake suite maintained and supported by Kitware (kitware.com/cmake).
```

The problem is that --stdlib can be ommited by design, while it's vital to have in compiler flags.

#7 - 31 May 2016 16:16 - T Modes

Does it help when you add
set(CMAKE_XCODE_ATTRIBUTE_CLANG_CXX_LIBRARY "libc++")
to CMake?

#8 - 09 Jun 2016 20:42 - Robin Mills

- Assignee deleted (Robin Mills)

#9 - 13 Jun 2016 11:47 - Robin Mills

- Assignee set to Robin Mills

- Target version changed from 1.0 to 0.28

#10 - 13 Jun 2016 11:47 - Robin Mills

- Status changed from New to Assigned

#11 - 14 Jul 2017 17:38 - Robin Mills

- Target version changed from 0.28 to 0.27

#12 - 18 Sep 2018 12:28 - Robin Mills

- Status changed from Assigned to New

- Assignee deleted (Robin Mills)

- Target version changed from 0.27 to 0.28

This is a top priority feature for v0.28.

#13 - 16 Nov 2018 09:19 - Robin Mills

- Status changed from New to Closed

- Assignee set to Robin Mills

- Target version changed from 0.28 to 0.27

- % Done changed from 0 to 100

- Estimated time set to 1.00 h

I'm closing this issue as part of the Exiv2 v0.27 Review process. The primary focus of Exiv2 v0.28 is to "modernise the code" to C++11 or later.

So, this issue is being closed on Redmine and will get a lot of attention in 2019.

Files

cxx11.diff	1.24 KB	31 May 2016	T Modes