Exiv2 - Bug #1032

CMake doesn't build svn_version.h anymore

15 Feb 2015 15:24 - Daniel Kaneider

Status:	Closed		**Start date:**	15 Feb 2015
Priority:	Normal		**Due date:**	
Assignee:	Robin Mills		**% Done:**	100%
Category:	build		**Estimated time:**	0.00 hour
Target version:	0.25			

Description

Hi,
as mentioned in the previous issue, commit 3479, caused that svn_version.h is not generated any more. CMake log output correctly reports the svn version, but 'exiv2.exe -V -v' just puts 0 (the default value). Not that I would particularly care about this bug, I just wanted to let you know.

Regards,
Daniel

Associated revisions

Revision 3610 - 24 Feb 2015 17:05 - Robin Mills

#1032. Thanks very much to Daniel for his help with this.

History

#1 - 16 Feb 2015 02:17 - Robin Mills

- Category set to build

- Status changed from New to Assigned

- Assignee set to Robin Mills

- Target version set to 0.25

#2 - 23 Feb 2015 18:17 - Robin Mills

Daniel

Can you help me with this, please? My aim is to generate the file src/svn_version.h as follows:

```
527 rmills@rmillsmbp:~/gnu/exiv2/trunk/src $ ./svn_version.sh
BASH_SOURCE=([0]="./svn_version.sh")
svn_version=3607
svn_version_h=svn_version.h
528 rmills@rmillsmbp:~/gnu/exiv2/trunk/src $ cat svn_version.h
#ifndef SVN_VERSION
#define SVN_VERSION 3607
#endif
```

There is a file include/exiv2/svn_version.h which is "hard-wired" to SVN_VERSION 0 and is the fallback to guarantee that #include "svn_version.h" will work and macro SVN_VERSION is defined.

I believe the bash script <exiv2dir>/src/svn_version.sh (called by the autotools) works fine. Last Fall, you reported issues with calling svn_version.sh using CMake/Command and provided a patch which I didn't understand. However, I respect you and therefore committed the patch to our repository.

The following code in <exiv2dir>/CMakeLists.txt to sets/reports the SVN revision number. It seems to work fine, although I don't know what Subversion_WC_INFO does.

```
FIND_PACKAGE(Subversion QUIET)
SET(SVN_REVISION "0")
IF(SUBVERSION_FOUND)
    IF(EXISTS "${CMAKE_SOURCE_DIR}/.svn")
        Subversion_WC_INFO(${PROJECT_SOURCE_DIR} svnExiv2)
        SET(SVN_REVISION "${svnExiv2_WC_REVISION}")
    ENDIF()
ENDIF()
MESSAGE(STATUS  "SVN version:  ${SVN_REVISION}")
```

There is a file <exiv2dir>/src/svn_version.h.in which seems to reference the environment string ${SVN_REVISION}. However, I don't understand how that file mutates to <exiv2dir>/src/svn_version.h.

```
525 rmills@rmillsmbp:~/gnu/exiv2/trunk $ svn log src/svn_version.h.in
------------------------------------------------------------------------
r3374 | robinwmills | 2014-10-11 16:54:56 +0100 (Sat, 11 Oct 2014) | 1 line

#994 Adding src/svn_version.h.in - apologies to Daniel for not committing this, and to Jehan for the inconveni
ence.
------------------------------------------------------------------------
526 rmills@rmillsmbp:~/gnu/exiv2/trunk $
```

I have many questions about CMake, however let's start by resolving this matter. I know you don't care about this particular issue, however we have to start somewhere. When we make progress with this issue, you will boost my confidence to tackle more serious concerns about MSVC/CMake.

#3 - 23 Feb 2015 20:51 - Daniel Kaneider

Alright.

The CMake installation ships with a set of Find* 3rd party modules. One of them deals with Subversion. On my windows machine, this is located under <cmake-dir>\share\cmake-3.2\Modules\FindSubversion.cmake. That is where Subversion_WC_INFO gets defined. It basically just calls svn binary to get some info about the current working copy. The other lines of code try to detect a subversion client silently, and to set the svn revision (if found) to a non-zero value.

svn_version.h is not supposed to exist in any source directory. It should rather be generated into the output directory, and copied (or CMake-installed) into the 'distribution' directory. I know this concept breaks some existing builds tools, especially msvc is a weird case. Some while ago I included in that patch

```
CONFIGURE_FILE( ${CMAKE_CURRENT_SOURCE_DIR}/svn_version.h.in
                ${CMAKE_CURRENT_BINARY_DIR}/svn_version.h @ONLY)
```

This works with the help of the template svn_version.h.in

```
#define SVN_VERSION @SVN_REVISION@
```

and replaces everything between the two -s. It's a CMake command which takes a template as input, and generates/replaces all found variables in the output file. And in order to include the header file for the compilation, I did

Again, the intention was to just have the .in@ template file in the source directory, but not the generated headers itself.

Especially important for VS (but extensible also for other build chains for compatibility) was a change in version.hpp, which didn't include the header file, if it wasn't generated. Using:

```
#if     defined(_MSC_VER) && !defined(CMAKE_BUILD_SVN)
 #define SVN_VERSION 0
#else
 #include "svn_version.h"
```

That last change isn't maybe the best approach, but the one which is working with the least changes. The other changes regarding CMake should be straightforward.

Summarizing to maintain compatibility as much as possible:
- consider svn_version.h in the source directory as deprecated, it should get generated. Moving it into a *compatibility* directory could be a first step.
- you could start to CMake-Configure that header file, by replacing it where it is right now

#4 - 23 Feb 2015 22:23 - Robin Mills

Bingo. I understand. The CONFIGURE_FILE/foo.h.in dance is the magic to make things work. Right, now I understand.

<exiv2dir>/src/svn_version.h is not in the code base. It is always generated. <exiv2dir>/include/exiv2/svn_version.h is there to save the day when src/svn_version.h is absent for any reason.

I'll progress this tomorrow and hope to submit. Then I'll look at [#1031](#1031)

Thank You very much for your help.

#5 - 28 Feb 2015 21:43 - Robin Mills

- Status changed from Assigned to Resolved

#6 - 08 May 2015 16:33 - Robin Mills

- % Done changed from 0 to 100

#7 - 21 Jun 2015 16:39 - Andreas Huggel

- Status changed from Resolved to Closed