

Exiv2 - Bug #1299

exiv2-0.26-trunk.tar.gz changed on download server

14 Jul 2017 14:20 - Jonathan Riddell

Status:	Closed	Start date:	14 Jul 2017
Priority:	Normal	Due date:	
Assignee:	Robin Mills	% Done:	100%
Category:	build	Estimated time:	1.00 hour
Target version:	0.27		

Description

The file at <http://www.exiv2.org/builds/exiv2-0.26-trunk.tar.gz> was changed on 7 July

The file was first released on 17 May. We package it in KDE neon and built it on 3 July. Rebuilding it today we find the file has changed.

It may have been compromised.

On <http://web.archive.org/web/20170606065325/http://www.exiv2.org/download.html> the md5sum is listed as f936d2ca5cbe1e18c71ca2baa5e84fb4

Currently on <http://www.exiv2.org/download.html> it is listed as 5399e3b570d7f9205f0e76d47582da4c

But the date has not changed

History

#1 - 14 Jul 2017 14:28 - Gilles Caulier

It's confirmed.

For digiKam bundles packaging, I already change 2 time in my scripts the checksums while downloading the official 0.26 tarball generated from trunk. Why ?

Current Exiv2 0.26 fail to build under Macport because the checksum is wrong !

A official tarball release must not be changed. A new tarball with a minor release ID must be published. This is the rules in Open Source...

#2 - 14 Jul 2017 15:21 - Robin Mills

- Category set to build

- Status changed from New to Closed

- Assignee set to Robin Mills

- Target version set to 0.28

Well spotted, Gentlemen. The file has not been compromised. The tar was regenerated without changing contents. The reason is documented here: <https://github.com/Exiv2/exiv2/issues/14>

The original tar was created on MacOS-X. The version of tar provided by Apple stores hidden files in the tar with names such as `._foo` which contain extended file-system metadata. When `$ tar -xf foo.tar.tz` is used on MacOS-X, the extended metadata is applied. However on Linux, these files appear on the filesystem and caused a CMake build failure on Linux when building the `i18n` `.po` files with the option `EXIV2_ENABLE_BUILD_PO`.

The only file involved is the source download. The other bundles with builds of Cygwin/MSVC/Linux/MacOS and MinGW have been created using tar on the target system.

For your information, I will start work on a dot release Exiv2 v0.26.1 when I return from vacation in Early August. I hope that will be ready in September. There will be no API changes in v0.26.1 and be a replacement for v0.26.0 However the build for v0.26.1 will be significantly changed as the Adobe XMPsdk will be an external library.

I am unaware of the "rules of open source". If you want this bundle to be relabelled as v0.26.1, I will change this.

#3 - 14 Jul 2017 15:36 - Gilles Caulier

0.26.0 => 0.26.1 ==> no API/ABI change. Only minor changes. See GNU guide :

<https://www.gnu.org/software/libtool/manual/libtool.html#Updating-version-info>

In 0.26.1 XMP sdk will be an external dependency. Right ?

You will manage this external dependency release ? Adobe licence permit to do it safety ?

#4 - 14 Jul 2017 16:04 - Robin Mills

I cannot yet say exactly how the dependency on Adobe XMPsdk will be handled as it's still a "work in progress".

I haven't understood your question concerning Adobe. Are you asking an engineering or a legal question?

From an Engineering point of view, It's likely that there will be a CMake option to download and build XMPsdk and will be enabled by default. So, it's likely that the users of CMake will notice very little change. Users of msvc will have to manually download Adobe XMPsdk and place it in a directory parallel to exiv2. This is similar to our current handling of expat and zlib. I haven't started to consider what changes are required by the Exiv2 autotools (./configure) build. It's likely that no changes are required, however the user will be responsible for building and installing XMPsdk.

From a legal point of view, I expect Adobe will be very happy about this. For the last 10 years, Exiv2 has contained a source copy of Adobe's code. As I didn't participate in the project to integrate Adobe's code with Exiv2, I can't say what (if anything) was modified. However as we build and link that code, it's not "pure Adobe". And I don't know if Adobe were consulted about distributing their code in this way.

In future, we will be building XMPsdk as released by Adobe without modification. We have tested Exiv2 with both the 2014 and 2016 Editions. Adobe license XMPsdk under BSD: <http://www.adobe.com/devnet/xmp/sdk/eula.html> I believe the way in which Exiv2 will use XMPsdk in future is legitimate.

#5 - 14 Jul 2017 16:27 - Robin Mills

- Target version changed from 0.28 to 0.27

- % Done changed from 0 to 100

- Estimated time set to 1.00 h

Gilles: You are 100% correct in saying that the checksum of source bundle of Exiv2 v0.26 has been changed twice since the release at the end of April. The most recent rebundling of the tar.gz was in July and that is what is being reported and discussed here. After the release, Andreas wrote and asked me to remove all offers to issue a Commercial License on the web-site and in various ReadMe files in the release. This was done in mid-May. As with the change in July, the code in the different bundles is identical.

#6 - 14 Jul 2017 22:26 - Gilles Caulier

Yes, i'm talking about legally point with Adobe licensing.

About technical point :

1/ a XMP sdk version already exist to be redistributed in Open Source world : libexempi

<https://github.com/hfiguiere/exempi>

2/ yes Cmake has all the capabilities to manage 3rd-party component to download and compile with a project. I use it to bundle the whole digiKam under Windows, MacOS, and Linux. Look here :

<https://cgit.kde.org/digikam-software-compilation.git/tree/project/bundles>

and especially the 3rdparty sub directory and all cmakeLists.txt script. The main cmake macro to know is ExternalProject_Add. This script download and compile Exiv2 for bundle packaging :

https://cgit.kde.org/digikam-software-compilation.git/tree/project/bundles/3rdparty/ext_exiv2/CMakeLists.txt

It's very simple and powerfull to use.

#7 - 15 Jul 2017 05:54 - Robin Mills

We've been asked by a user to support "pure external Adobe XMPsdk" as an external library. This enables host applications to link and use XMPsdk without encountering linking errors. Currently, static externals in <exiv2-dir>/xmpsdk/src conflict those used by Adobe. Nobody has requested that we support exempi, although once XMPsdk is an external library, the use of exempi may be an alternative way to build Adobe's code.

I don't see any licensing issue with Adobe and our plans. Can you see a conflict that I have overlooked?

Thank you for the pointers to your scripts concerning CMake/ExternalProject_Add That's what I'm planning to use and why I said:

From an Engineering point of view, It's likely that there will be a CMake option to download and build XMPsdk and will be enabled by default.

The most important issue concerning the Adobe XMP is the linking issue. I suggested to the user that the static externals in <exiv2dir>/xmpsdk/src be wrapped in the Exiv2 namespace to avoid conflict. The user was able to construct something along those lines, however he was unhappy and requested the "pure external Adobe XMPsdk" approach. The second objective is to keep our XMP code up-to-date as it has not been refreshed for about 10 years. The XMPsdk support destined for v0.26.1 enables the user to choose either the 2016 or 2014 Editions from Adobe.

I will be happy to hear your thoughts about this if you think this approach is wrong, illegal, or won't work. I've explained the rationale and hope that v0.26.1 can ship in September and am working hard to make this happen. If you believe there is a flaw in what is being done, it's not too late to

consider alternatives.

I hope that v0.26.1 will also include improved Video/Read support.

#8 - 15 Jul 2017 06:19 - Gilles Caulier

I don't see any legal issue with Adobe SDK, if licensing from Adobe do not change since a while. I'm not expert here... If Exempi library, based on Adobe SDK, is updated with recent release from Adobe code, well, all must be fine.

Note : Exempi, as i know include compilation fixes for portabilities as Adobe do not report as UPSTREAM. This can prevent any patch to include in your side in the future (so less work to maintain portability). Look well Exempi project description.

<https://libopenraw.freedesktop.org/wiki/Exempi/>

Another advantage is to see Exempi lib already included in Linux distributions.

About the linking issue with Exiv2 and XMP sdk, yes, in the pass digiKam has seen some runtime issue due to wrong linking rules, fixed with cmake > 3.0 options as i remember.