

Exiv2 - Bug #1173

Shared library on Mac OS X has incorrect name

31 Mar 2016 02:01 - Ilya Kulakov

Status:	Closed	Start date:	31 Mar 2016
Priority:	Normal	Due date:	
Assignee:	Robin Mills	% Done:	100%
Category:	build	Estimated time:	1.00 hour
Target version:	0.26		
Description			
For some reason, install name of the shared library (e.g. returned by `otool -D`) uses name of the SONAME (e.g. libexiv2.14.dylib) instead of a full name (e.g. libexiv2.14.0.0.dylib).			
This is incorrect, because install name should match real name of the library, not a name of its SONAME symlink.			

History

#1 - 31 Mar 2016 09:21 - Robin Mills

- Status changed from New to Assigned
- Target version set to 0.26
- % Done changed from 0 to 30
- Estimated time set to 2.00 h

We're going to have to discuss this a little more to be "on the same page". So, to help me decide what to do, may I ask the question. **Why are you concerned about this? Is there a real problem to be solved?**

I have built and installed exiv2 with the autotools and CMake. They behave a differently.

Let's define how libraries/install_name and apps should be installed and linked. I have looked at curl as installed by Apple in /usr/lib.

```
513 rmills@rmillsmbp:/usr/lib $ ls -alt | grep curl
-rwxr-xr-x  1 root  wheel   785312 12 Mar 08:36 libcurl.4.dylib
lrwxr-xr-x  1 root  wheel     15 1 Oct 13:47 libcurl.3.dylib -> libcurl.4.dylib
lrwxr-xr-x  1 root  wheel     15 1 Oct 13:47 libcurl.dylib -> libcurl.4.dylib
514 rmills@rmillsmbp:/usr/lib $ otool -L libcurl.4.dylib
libcurl.4.dylib:
  /usr/lib/libcurl.4.dylib (compatibility version 7.0.0, current version 8.0.0)
  /System/Library/Frameworks/Security.framework/Versions/A/Security (compatibility version 1.0.0, current ve
rsion 57332.0.0)
  /System/Library/Frameworks/CoreFoundation.framework/Versions/A/CoreFoundation (compatibility version 150.0
.0, current version 1231.0.0)
  /System/Library/Frameworks/LDAP.framework/Versions/A/LDAP (compatibility version 1.0.0, current version 2.
4.0)
  /System/Library/Frameworks/Kerberos.framework/Versions/A/Kerberos (compatibility version 5.0.0, current ve
rsion 6.0.0)
  /usr/lib/libz.1.dylib (compatibility version 1.0.0, current version 1.2.5)
  /usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1225.0.0)
515 rmills@rmillsmbp:/usr/lib $ otool -L ../bin/curl
../bin/curl:
  /usr/lib/libcurl.4.dylib (compatibility version 7.0.0, current version 8.0.0)
  /usr/lib/libz.1.dylib (compatibility version 1.0.0, current version 1.2.5)
  /usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1225.0.0)
516 rmills@rmillsmbp:/usr/lib $
```

When I build exiv2 with the autotools (./configure), we replicate this pattern of files/links and install_name.

```
522 rmills@rmillsmbp:/usr/local/lib $ ls -alt *exiv2*
-rw-r--r--  1 root  admin  2470136 31 Mar 09:43 libexiv2.14.dylib
-rw-r--r--  1 root  admin  5086480 31 Mar 09:43 libexiv2.a
lrwxr-xr-x  1 root  admin    17 31 Mar 09:43 libexiv2.dylib -> libexiv2.14.dylib
-rw-r--r--  1 root  admin   1088 31 Mar 09:43 libexiv2.la
523 rmills@rmillsmbp:/usr/local/lib $ otool -L libexiv2.14.dylib
libexiv2.14.dylib:
  /usr/local/lib/libexiv2.14.dylib (compatibility version 15.0.0, current version 15.0.0)
  /usr/local/lib/libintl.8.dylib (compatibility version 10.0.0, current version 10.2.0)
```

```

/usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1226.10.1)
/usr/lib/libiconv.2.dylib (compatibility version 7.0.0, current version 7.0.0)
/usr/local/lib/libz.1.dylib (compatibility version 1.0.0, current version 1.2.8)
/usr/local/lib/libexpat.1.dylib (compatibility version 8.0.0, current version 8.0.0)
/System/Library/Frameworks/CoreFoundation.framework/Versions/A/CoreFoundation (compatibility version 150.0
.0, current version 1258.1.0)
/usr/lib/libc++.1.dylib (compatibility version 1.0.0, current version 120.1.0)
524 rmills@rmillsmbp:/usr/local/lib $ otool -L ../bin/exiv2
../bin/exiv2:
/usr/local/lib/libexiv2.14.dylib (compatibility version 15.0.0, current version 15.0.0)
/usr/local/lib/libintl.8.dylib (compatibility version 10.0.0, current version 10.2.0)
/usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1226.10.1)
/usr/lib/libiconv.2.dylib (compatibility version 7.0.0, current version 7.0.0)
/usr/local/lib/libz.1.dylib (compatibility version 1.0.0, current version 1.2.8)
/usr/local/lib/libexpat.1.dylib (compatibility version 8.0.0, current version 8.0.0)
/usr/lib/libc++.1.dylib (compatibility version 1.0.0, current version 120.1.0)
525 rmills@rmillsmbp:/usr/local/lib $

```

When I build and install exiv2 with CMake, I see:

```

591 rmills@rmillsmbp:/usr/local/lib $ ls -alt *exiv2*
lrwxr-xr-x  1 root  admin      21 31 Mar 05:00 libexiv2.14.dylib -> libexiv2.14.0.0.dylib
lrwxr-xr-x  1 root  admin      17 31 Mar 05:00 libexiv2.dylib -> libexiv2.14.dylib
-rwxr-xr-x  1 root  admin  4779248 31 Mar 04:52 libexiv2.14.0.0.dylib
592 rmills@rmillsmbp:/usr/local/lib $ otool -L libexiv2.14.0.0.dylib
libexiv2.14.0.0.dylib:
@rpath/libexiv2.14.dylib (compatibility version 14.0.0, current version 14.0.0)
/usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1226.10.1)
/opt/local/lib/libexpat.1.dylib (compatibility version 8.0.0, current version 8.0.0)
/opt/local/lib/libz.1.dylib (compatibility version 1.0.0, current version 1.2.8)
/opt/local/lib/libiconv.2.dylib (compatibility version 8.0.0, current version 8.1.0)
/usr/lib/libc++.1.dylib (compatibility version 1.0.0, current version 120.1.0)
593 rmills@rmillsmbp:/usr/local/lib $ otool -L ../bin/exiv2
../bin/exiv2:
@rpath/libexiv2.14.dylib (compatibility version 14.0.0, current version 14.0.0)
/usr/lib/libc++.1.dylib (compatibility version 1.0.0, current version 120.1.0)
/usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1226.10.1)
594 rmills@rmillsmbp:/usr/local/lib $

```

He looks a bit confused to me. The install_name in /usr/local/bin/exiv2 has been correctly copied from the library: **@rpath/libexiv2.14.dylib** The issue is that the install_name in the library itself is wrong. libexiv2.14.dylib is a link and not the library itself.

So can we agree that this issue only applies to the CMake build? There are two possible fixes:

1. He should have created the library in the file /usr/local/lib/libexiv2.14.dylib and created a symbolic link to that file from /usr/local/lib/libexiv2.14.0.0.dylib. This would replicate the Apple/curl pattern.
2. He can create the library in the file /usr/local/lib/libexiv2.14.0.0.dylib, however the install_name in that library should be **@rpath/libexiv2.14.0.0.dylib**

Curiously, I notice that CMake has installed the library with the wrong time stamp. It's off by 7 hours! He's determined to be in Cupertino or something which is 8 hours from my machine in England and convoluted by day-light saving or something.

I feel this might be a bug in CMake. I recall that a user requested a change to our CMake/rpath support. When I make a change to our CMake files to give you happiness, this will cause somebody else to complain. So that brings me back to the questions: **Why are you concerned about this? Is there a real problem to be solved?**

#2 - 01 Apr 2016 17:31 - Ilya Kulakov

Robin,

You're perfectly correct that the issue only exists in CMake build.

1. I think the name of the library should be libexiv2.14.0.0.dylib because it's a full name and libexiv2.14.dylib should be a symlink which may optionally point to something like libexiv2.14.0.1.dylib or libexiv2.14.1.0.dylib. The reasoning is that symlink with compatibility name (SONAME) should point to the actual version, but libraries themselves should exist under their real names. Otherwise you simply cannot have multiple compatible versions of the same library installed simultaneously
2. The install name of the library should indeed be real name of library's file

I'm concerned about this, because install_name must match real name of the library, otherwise when you link against this library, wrong library identifier will be embedded in the binary.

#3 - 01 Apr 2016 17:35 - Ilya Kulakov

Actually, I take my words back: it's not required: many libraries in /usr/lib does not do that.

Moreover, the behavior is rather inconsistent. Looks like it can be any way, and it's up to a developer to choose how to do that.

#4 - 01 Apr 2016 18:22 - Robin Mills

- *Status changed from Assigned to Closed*
- *% Done changed from 30 to 100*
- *Estimated time changed from 2.00 h to 1.00 h*

Thanks for getting me back to me about this. I could easily add something to CMake to execute **install_name_tool** to fix this (to conform with option 1 above). I'm in the last few weeks before code complete for v0.26. At this stage in the release cycle I prefer to only submit essential changes. So, I'm going to close this.

I wonder if this is a bug in CMake. If you have time, maybe you could investigate with a simple "hello world" application and "hello world" library.