

Exiv2 - Bug #1293

MacOS-X CMake/autotools incompatibility handling of -compatibility_version

26 Apr 2017 16:26 - Robin Mills

Status:	Closed	Start date:	26 Apr 2017
Priority:	Normal	Due date:	
Assignee:	Robin Mills	% Done:	100%
Category:	build	Estimated time:	2.00 hours
Target version:	0.26		

Description

[r4769](#)

CMake libraries are not compatible with Autotools libraries

Autotools libraries have a "compatibility version" of 16.

```
750 rmills@rmillsmbp:~/gnu/exiv2/trunk $ otool -L /usr/local/lib/libexiv2.15.dylib
/usr/local/lib/libexiv2.15.dylib:
  /usr/local/lib/libexiv2.15.dylib (compatibility version 16.0.0, current version 16.0.0)
  /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 1238.51.1)
  /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 v
ersion 150.0.0, current version 1349.65.0)
  /usr/lib/libc++.1.dylib (compatibility version 1.0.0, current version 307.5.0)
751 rmills@rmillsmbp:~/gnu/exiv2/trunk $
```

CMake builds have compatibility set to 15.

```
755 rmills@rmillsmbp:~/gnu/exiv2/cmakebuild/build $ otool -L /usr/local/lib/libexiv2.15.dylib
/usr/local/lib/libexiv2.15.dylib:
  @rpath/libexiv2.15.dylib (compatibility version 15.0.0, current version 15.0.0)
  /usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1238.51.1)
  /usr/local/lib/libexpat.1.dylib (compatibility version 8.0.0, current version 8.0.0)
  @loader_path/libz.1.2.8.dylib (compatibility version 1.0.0, current version 1.2.8)
  /usr/lib/libiconv.2.dylib (compatibility version 7.0.0, current version 7.0.0)
  /usr/lib/libc++.1.dylib (compatibility version 1.0.0, current version 307.5.0)
756 rmills@rmillsmbp:~/gnu/exiv2/cmakebuild/build $
```

This issue is created by the following code in config/ltmain.sh:

```
darwin)
  # Like Linux, but with the current version available in
  # verstring for coding it into the library header
  func_arith $current - $age
  major=.$func_arith_result
  versuffix="$major.$age.$revision"
  # Darwin ld doesn't like 0 for these options...
  func_arith $current + 1
  minor_current=$func_arith_result
  xlcverstring="{wl}-compatibility_version ${wl}$minor_current ${wl}-current_version ${wl}$mi
nor_current.$revision"
  verstring="-compatibility_version $minor_current -current_version $minor_current.$revision"
  ;;
```

The "compatibility version" in Exiv2/autotools v0.25 is 15 and the library version is 14.

Having the library file of version v0.25 and the file as libexiv2.14.dylib and "compatibility version" 15 is confusing. To rationalise, I will sync everything at 26 for v0.26. For v0.26, the shared library will be libexiv2.26.dylib and the compatibility version will be 26.

The consequence of this is that apps build with autotools may fail to find the correct library. I discovered this while trying to execute the test harness on a mix of CMake libraries and Autotools applications (such as bin/exifprint).

Another difference between CMake and Autotools is the use of @rpath and @loader_path by CMake. Both are correct. I prefer CMake's use of @rpath as the libraries can be easily moved to another directory. The test suite takes advantage of this to run tests without installing libraries in /usr/local/lib. Autotools enforces the absolute path from --prefix in built libraries.