CrystaX NDK - Feature #898

Add openssl as a prebuilt library

02/09/2015 11:51 AM - Alex Afanasyev

Status:	Open	**Start date:**	02/09/2015
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
CPU Architecture:		**Toolchain:**	
Host OS:		**Android version:**	

Description

I'm not sure about this one, but I believe android is shipped with old version of openssl library. At least in several NDK documentations I saw mentioning of compiling newer version of openssl. Personally I'm not using it (not yet/not anymore), but it could be useful for many other projects.

History

#1 - 02/09/2015 02:37 PM - Dmitry Moskalchuk

Alex Afanasyev wrote:

> I'm not sure about this one, but I believe android is shipped with old version of openssl library. At least in several NDK documentations I saw mentioning of compiling newer version of openssl. Personally I'm not using it (not yet/not anymore), but it could be useful for many other projects.

NDK don't include OpenSSL at all. Android itself is shipped with openssl library, but it's not available for applications developers. Strictly speaking, there is no technical limitations since dlopen() is still there; however, no guarantees regarding version of the openssl, regarding location of binaries, it's names etc. All this lead to the very unstable and error-prone behaviour if app start using system-installed openssl.

To be able use it properly, one should use own openssl library, packaged with application, and this is what we'd like to to - provide prebuilt openssl binaries as part of NDK.

#2 - 08/28/2015 07:58 PM - Dmitry Moskalchuk

- Tracker changed from Bug to Feature

- Subject changed from Add openssl as a prebuild library to Add openssl as a prebuilt library

#3 - 04/18/2017 10:38 PM - Boris Skegin

Would be nice if you could provide *libssh* as well.

This https://www.libssh.org/archive/libssh/2015-11/0000006.html might help, I tested the workflow with android-ndkr10 and API level 21.

#4 - 11/13/2017 11:28 PM - Alex Afanasyev

I see that openssl 1.0.2k is bundled with the nightly build of CrystaX (b1028), but there is interesting issue:

- the bundled version has incorrect soname for the .so files

```
/opt/crystax-ndk-b1028/packages/openssl/1.0.2k/libs/armeabi-v7a $ /opt/android-sdk/ndk-bundle/ndk-depends libssl.so
WARNING: Library has invalid soname ('libssl.so.1.0.0'): libssl.so
WARNING: Could not find library: libcrypto.so.1.0.0
WARNING: Could not find library: libcrystax.so
libssl.so
libdl.so
libcrystax.so
libcrypto.so.1.0.0
```

However, I compiled 1.0.2m version using the provided script build/instruments/build-target-openssl.sh and it has correct sonames:

```
/opt/crystax-ndk-b1028/packages/openssl/1.0.2m/libs/armeabi-v7a $ /opt/android-sdk/ndk-bundle/ndk-depends libssl.so
WARNING: Could not find library: libcrystax.so
```

```
libssl.so
libcrypto.so
libdl.so
libcrystax.so
```

I used version downloaded from openssl website. There was an issue with version detection that required a small update of the regex. The following worked for me (difference in removed \?, couldn't convince my sed to work with it):

```
$(cat $OPENSSL_SRCDIR/crypto/opensslv.h | sed -n 's/#[ \t]*define[ \t]*OPENSSL_VERSION_TEXT[ \t]*"OpenSSL[ \t]*\([0-9\.]*[A-Za-z]\)[A-Za-z0-9 \.]*"/\1/p')
```