wolfSSL JNI Manual

April 24, 2014, version 1.0.2

Introduction

The wolfSSL JNI package provides a Java interface to the wolfSSL SSL/TLS library, providing Java applications with SSL/TLS support up to the current TLS 1.2 and DTLS 1.2 standards. The interface is provided through the use of JNI and standard Java practices. In addition to the interface, the package provides an example client and server, written in Java, which utilize the interface to make an SSL/TLS connection.

As this interface wraps around the native wolfSSL (CyaSSL) library, this document should be used together with the CyaSSL Manual (http://yassl.com/yaSSL/Docs-cyassl-manual-toc.html).

Table of Contents

Copyright 2014 wolfSSL Inc. All rights reserved.

Getting wolfSSL JNI Source Code

The most recent version of wolfSSL JNI can be downloaded from the wolfSSL website as a ZIP file from the Download page:

http://yassl.com/yaSSL/download/downloadForm.php

Requirements

To compile and use this interface, users must have Java installed on the development machine. Prior to compiling the JNI wrapper, the wolfSSL (CyaSSL) library must be compiled and installed in a location which can be found by the JNI wrapper. This release of the Java wrapper has been tested against CyaSSL 2.9.4.

The package utilizes the ant build system to make compilation, testing, and documentation easier and more streamlined. As such, if users wish to use the existing ant build script, ant will need to be installed and available on the development machine.

JUnit is also used for unit tests. JUnit will need to be installed on the development machine in order for unit tests to work. Users can download JUnit from http://www.junit.org.

Package Design

The Java interface package has the following directory structure.

<package_root>

build.xml	(ant build script)
/docs	(javadocs)
/examples	(examples)
/certs	(test certs/keys)
java.sh	(JNI gcc script)
/lib	(output directory for compiled JNI lib and JAR file)
/native	(native JNI code)
/src	
/java	(Java sources)
/test	(test code)

Building

CyaSSL will need to be compiled and installed on the development machine using the following build options:

Copyright 2014 wolfSSL Inc. All rights reserved.

```
cd cyassl-2.9.4

./configure --enable-dtls --enable-opensslextra --enable-crl --enable-ocsp
--enable-crl-monitor --enable-savesession --enable-savecert
--enable-atomicuser --enable-ecc --enable-pkcallbacks

make
sudo make install
```

To compile the JNI interface, run the following commands from the root package directory.

`./java.sh`	(compiles native JNI sources into shared library)
`ant`	(compiles Java sources, javadocs, JNI headers, JAR, and runs available tests)

The java.sh script assumes that Java is installed at the following locations. If your system differs from the below paths, please modify java.sh to match your environment before running the script.

OS X:	/System/Library/Frameworks/JavaVM.framework/Headers
Linux:	/usr/lib/jvm/java-6-openjdk/include

Cleanup

To clean the package, run the following commands from the root directory:

`ant clean`	(cleans all Java sources and .JAR)
`ant cleanjni`	(cleans native object files and shared library)

Examples

This interface includes an example server and client written in Java to provide developers with a simple usage example. The example client and server are located in the <package_root>/examples directory. Each is accompanied by a wrapper script. The wrapper script should be used when running the programs to more easily set desired classpath and library path options.

The examples can be run from the root package directory using:

```
./examples/server.sh
./examples/client.sh
```

Each example provides several command line options which can be used to customize how the example(s) run. Use the “-?” option to show available command line options:

Copyright 2014 wolfSSL Inc. All rights reserved.

```
./examples/server.sh -?
Java example server usage:
-?          Help, print this usage
-p <num>    Port to connect to, default 11111
-v <num>    SSL version [0-3], SSLv3(0) - TLS1.2(3)), default 3
-l <str>    Cipher list
-c <file>   Certificate file,       default ../certs/client-cert.pem
-k <file>   Key file,               default ../certs/client-key.pem
-A <file>   Certificate Authority file,    default ../certs/client-cert.pem
-d          Disable peer checks
-iocb       Enable test I/O callbacks
-logtest    Enable test logging callback
-U          Atomic User Record Layer Callbacks
-P          Public Key Callbacks
```

The examples demonstrate usage of several things, including using custom I/O callbacks, a custom verify callback, DTLS cookie generation callback, Atomic Record callbacks, and public key (ECC, RSA) callbacks.

Sources for the example client can be found in <package_root>/examples/Client.java and the example server in <package_root>/examples/Server.java. Custom I/O callbacks can be found in MyRecvCallback.java and MySendCallback.java, with the I/O context found in MyIOContext.java. The custom verify callback can be found in VerifyCallback.java. Examples of atomic user record layer callbacks and public key callbacks are also included in the <package_root>/examples directory.

The verify callback currently returns 1 (failure), but does no processing of the peer certificate chain. A real-world application would want to do any desired certificate verification processing before returning failure or success.

Currently DTLS usage requires custom I/O callbacks and DTLS cookie generation callback to be registered. Examples of these can be found in Server.java and Client.java.

API Documentation

The ant build system compiles up-to-date Javadocs when run. To view the most current Javadocs for the package, open the local javadocs index file in a web browser.

<package_root>/docs/index.html

Copyright 2014 wolfSSL Inc. All rights reserved.

or browse to the online version at:

http://www.wolfssl.com/documentation/wolfssl-jni-javadocs/index.html

License

Like wolfSSL (CyaSSL), this package is dual licensed under both the GPLv2 as well as a standard commercial license. Full license details can be found on the wolfSSL Licensing page. The GPLv2 license header included in the wolfSSL JNI download package is copied below:

```
 * Copyright (C) 2006-2014 wolfSSL Inc.
 *
 * This file is part of CyaSSL.
 *
 * CyaSSL is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * CyaSSL is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA
```

Support

Please send questions or comments to wolfSSL at **support@wolfssl.com**.

References

CyaSSL Product Page: http://yassl.com/yaSSL/Products-cyassl.html
CyaSSL Documentation: http://yassl.com/yaSSL/Docs.html
CyaSSL Manual: http://yassl.com/yaSSL/Docs-cyassl-manual-toc.html

Copyright 2014 wolfSSL Inc. All rights reserved.