Building Tomcat on MacOS

Building Tomcat from source on MacOS can require some additional work, especially if you want to build all of the optional items such as `libtcnative`.

Building Tomcat

Building Tomcat itself is fairly straightforward. Simply download the source distribution of Tomcat and follow the instructions in the BUILDING.txt file bundled with the distribution.

Briefly, you'll need:

1. The Tomcat source tarball (the ZIP file is fine, but we're on UNIX, so the tarball is more natural)
2. A Java Development Kit, available from Java.net
3. Apache ant, available from the Apache ant downloads page

Once you have all that,

```
ant deploy
```

You may have to set your `JAVA_HOME` environment variable and/or specify the full path to your `ant` binary.

Building libtcnative

Building `libtcnative` it fairly straightforward as well, but you will need a number of prerequisites that are not terribly obvious as to how to get them.

1. Xcode command-line tools, available from Apple's developer tools download site. Make sure you get the proper version for your XCode version (if you have XCode already installed) and your OS version (10.x)
2. Apache Portal Runtime (APR), available either directly from Apache (APR downloads) or by using https://brew.sh/ (`brew install apr`) 1. (Optional) OpenSSL 1.1.1 (latest at the time of this writing), available either directly from OpenSSL (OpenSSL Downloads) or by using brew ({{{brew install openssl1

Download and unpack the libtcnative sources:

```
$ tar xzf tomcat-native-x.y.x.tar.gz
$ cd tomcat-native-x.y.z/native
```

Using OpenSSL

If using OpenSSL:

```
$ ./configure --with-ssl=[path to OpenSSL] --with-apr=[path to APR] --with-java-home=[your java home]
```

For example:

```
$ ./configure --with-ssl=/usr/local/Cellar/openssl@1.1/1.1.1 --with-apr=/usr/local/Cellar/apr/1.6.5 --with-java-home=/Library/Java/JavaVirtualMachines/jdk1.8.0_181.jdk/Contents/Home
```

Without OpenSSL (LibreSSL)

If not using OpenSSL:

```
./configure --with-ssl=yes --with-apr=[path to APR] --with-java-home=[your java home]
```

```
./configure --with-ssl=yes --with-apr=/usr/local/Cellar/apr/1.6.5 --with-java-home=/Library/Java/JavaVirtualMachines/jdk1.8.0_181.jdk/Contents/Home
```

In either case (OpenSSL or not), proceed with the build process:

```
make
```

Once this process has completed, your built libraries can be found in `.libs/`.

Installing libtcnative

While you can set `ld.library.path` to include libraries from all over the place, I recommend that you copy everything into one place so you can easily find everything and it won't interfere with anything else on your system.

First, copy the `libtcnative` binaries from where they were built:

```
$ cp -aR tomcat-native-x.y.z/native/.libs/* apache-tomcat-x.y.z-src/output/build/bin/
```

Next, copy the APR libraries:

```
$ cp -aR $APR_HOME/libexec/lib/* apache-tomcat-x.y.z-src/output/build/bin/
```

Finally, if you are using a custom OpenSSL build, copy those libraries as well:

```
$ cp -aR $OPENSSL_HOME/lib/* apache-tomcat-x.y.z-src/output/build/bin/
```

Running Tomcat with libtcnative

Tomcat (really Java) needs to know where to find these native libraries. We do that by setting the `java.library.path` environment variable for the JVM during startup. The easiest way to do this is by setting `CATALINA_OPTS` on startup. This can be done by adding this line to `bin/setenv.sh`:

```
export CATALINA_OPTS="-Djava.library.path=$CATALINA_HOME/bin"
```

Then you can startup Tomcat as usual, either:

```
$ bin/startup.sh
```

or

```
$ bin/catalina.sh start
```