

# Table of Contents

<b>Building Client-Side Libraries.....</b>	<b>1</b>
Building Java library.....	1
Building Python library.....	1
Building C++ library.....	1

# Building Client-Side Libraries

When you're preparing new Tudas release, you have to build all modules and transform them into the form that is comfortable for end-user. This process varies depending on programming language. In main Tudas project directory you can find release directory. It should be updated each time you want to provide changes in Tudas libraries and shared to Tudas users. The details are described below.

## Building Java library

Tudas java library is generated automatically by Maven. To build your current state, invoke command (from main Tudas project directory):

```
mvn package
```

The result jar is put into *clients/javaClient/target* directory. You have to move this file into *release/java/lib* directory:

```
mv -f ./clients/javaClient/target/javaClient-x.x.x-with-all-dependencies ./release/java/lib/tudas
```

## Building Python library

Thanks to Python's flexible nature, there is no need of compiling any files. All you have to do is replace old version of python code with new one in release directory:

```
rm -rf ./release/python/lib/tudas
cp -r ./clients/pythonClient/tudas ./release/python/lib
```

## Building C++ library

Final c++ shared library is compiled only for specific platform and architecture. That's why you have to perform build steps on Scientific Linux CERN if it should work on CERN machines (and that's our default approach).

1. Ensure, that you have built additional c++ libraries. It is performed automatically first time you call *maven compile* command (and takes a lot of time). If libraries have been built correctly, there should be appropriate content in *./clients/cppClient/lib* and *./clients/cppClient/include* directories.

2. Change your directory to c++ source directory:

```
cd ./clients/cppClient/src/main/cpp
```

3. Now you can build shared library file:

```
g++ -I./include -I./generated -I../../../../include -fPIC -c *.cpp generated/*.cpp
g++ -shared -L../../../../lib -lIce -lIceUtil -llog4cpp -lPocoFoundation -lPocoNet -o libtudas.so *
```

4. Remove generated obj files:

```
rm -f *.o
```

5. Move your library file into the release directory:

```
mv ./clients/cppClient/src/main/cpp/libtudas.so ./release/c++/lib/
```

This topic: TOTEM > CompDBTudasBuildingClientsLibraries

Topic revision: r1 - 2012-09-12 - unknown



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback