Building the Scripted Display Tools (sdt/sdt3d)

Table of Contents

1. Building sdt3d

The sdt3d application has external dependencies on NASA's Worldwind JDK, joglutils (a library that provides Java bindings for OpenGL and 3d Model support), and protolib-jni (a library that provides java native interface to NRL's protolib protoPipe implementation). Zip files of the Worldwind JDK and joglutils distributions known to work with the latest sdt3d code base are posted on the Protean Forge proteantools project release site in the latest sdt file release directory. The protolib-jni svn repository is also hosted on the Protean Forge website in the protolib project. See the instructions specific to your OS for further build instructions.

Those without developer access can find the latest open source version of the NASA Worldwind JDK at http://worldwind.arc.nasa.gov/java/index.html and the protolib source at http://downloads.pf.itd.nrl.navy.mil/protolib/. The location of the joglutils library is unreliable at best but the latest source may be found at: https://github.com/sgothel/jogl-utils/.

1.1. Linux Build

For Linux, a Makefile is provided that will build a sdt3d.jar file: (Note that you must have JAVA_HOME defined and javac and jar available in your path). First download the latest worldwind and joglutils zip files available in the latest sdt file release directory. Unzip the release in the sdt/makefiles directory and ensure the zip files are named joglutils.zip and worldwind.zip. From the makefiles directory make sdt3d. "make -f Makefile all" will make sdt, sdtcmd, and sdt3d.

```
cd sdt/makefiles
make -f Makefile.linux sdt3d
```

The sdt3d.sh helper script in the makefiles directory can then be used to launch the sdt3d jar file.

```
./sdt3d.sh
```

1.2. OSX Build

You may build sdt3d under osx via the makefile script described for a linux build or via Eclipse as described in the windows build section.

If protolib-jni does not build under osx it may be necessary to manually symlink to the include directory as appropriate for your jdk version, e.g.

```
cd /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/
sudo ln -s /System/Library/Frameworks/JavaVM.framework/Headers include
```

1.3. Windows Build

The easiest way to build sdt3d under windows is to use the Eclipse tool with the following four projects:

1. sdt3d

   Within Eclipse, check the sdt3d java src files out of the sdt SVN repository. (You will need developer access to the sdt project hosted on the protean forge website)

   ```
   SVN root = "/svnroot/proteantools/trunk/sdt"
   ```

   Check out the /src/java folder into a new project and name it sdt3d. Documentation and examples are available in the /sdt/doc and /sdt/examples folders that you may load into a separate project in your workspace.

   If you do not have developer access you may download the sdt nightly build distribution from http://downloads.pf.itd.nrl.navy.mil/sdt/ and load the sdt3d source files directly from sdt/src/java.

   The java folder contains Eclipse .project and .classpath configuration files that should be used to build the sdt3d project. (They expect the worldwind, joglutils, and protolib-jni project directories to be relative to the sdt3d as described in the following steps).

   Modify the sdt3d java build path to find the java jre libraries in the appropriate location for your system if necessary. (See the sdt3d project properties java build path libraries section)

2. protolib-jni

   If you have developer access check the protolib-jni file out of SVN directly into Eclipse:

   ```
   SVN root = "/svnroot/protolib"
   ```

   Check out the /trunk/src/java folder and name it protolib-jni.

   If you do not have developer access, get the nightly build distribution from http://downloads.pf.itd.nrl.navy.mil/protolib/ and load the source files directly from /protolib/src/java.

   Protolib-jni dll, jar, and library files are provided the src/java directory for your convenience and are also available in the protolib binary distribution. They should be loaded into your protolib project.

   If you have need to build protolib-jni, use a Visual Studio CMD (so that you have the correct environment variables set) and change directory to the top level protolib directory. Clean the distribution and configure waf to build the protolib-jni. Finally run waf to create the protolib-jni dll, jar, and library files in the build/default directory.

   ```
   waf distclean
   waf configure --disable-wx --disable-vif --disable-manet --static --build-java
   waf
   ```

3. joglutils

   Download the joglutils zip file from the latest sdt3d file release on the protean forge website. Unzip the project relative to the sdt3d project in your Eclipse workspace directory. Load into the Eclipse workspace using the build.xml file provided in the joglutils make directory. This project should be relative to the sdt3d project and be called joglutils.

   Modify the joglutils java build path to find the java jre (rt.jar) library in the appropriate location for your system. (See the joglutils project properties java build path libraries section).

4. worldwind

Download the worldwind zip file in the latest sdt file release on the protean forge website. Unzip the project relative to the sdt3d project in your Eclipse workspace directory. Load into the Eclipse workspace using the build.xml ant file provided in the worldwind directory.

Add the jogl.jar file in the worldwind workspace to the joglutils project. (See the joglutils project properties java build path libraries section).

2. Building sdt and sdtcmd

To build the sdt application, you will need the wxWidgets library for your platform freely-available from:

http://www.wxwidgets.org

The current release depends on wxWidgets 2.9 or wxWidgets 2.8. Previous releases (1.1a1 and above) use wxWidgets 2.6.x. sdt releases 1.0a1-1.0a8 use wxWidgets 2.4.x.

Please note that binary releases of wxWidgets do not include the header files necessary to compile sdt, so you must download and build the source code. Please see the wxWidgets readme for instructions. It is recommended to compile wxWidgets using the "--disable-shared" config option so that the binary may be moved to machines that do not have wxWidgets installed.

2.1. Windows Build

Sdt src code is available on NRL's protean forge web site at:

```
SVN root = "/svnroot/proteantools/trunk/sdt"
```

If you do not have developer access you may download the binary distribution from http://downloads.pf.itd.nrl.navy.mil/sdt/

Microsoft Visual Studio project files are available in the distribution in the makefiles/win32 directory. Load the sdt solution file into Microsoft Visual Studio. Project files are available that build sdt and sdtcmd.

2.2. Linux Build

Sdt src code is available on NRL's protean forge web site at:

```
SVN root = "/svnroot/proteantools/trunk/sdt"
```

If you do not have developer access you may download the binary distribution from http://downloads.pf.itd.nrl.navy.mil/sdt/

From the makefiles directory make sdt and sdtcmd. "make -f Makefile all" will make sdt,sdtcmd, and sdt3d.

```
cd sdt/makefiles
make -f Makefile sdt
make -f Makefile sdtcmd
```