



Building the Scripted Display Tools (sdt/sdt3d)

Table of Contents

1. Building sdt3d	1
1.1. Linux Build	1
1.2. OSX Build	2
1.3. Windows Build	2
2. Building sdt and sdtcmd	4
2.1. Windows Build	4
2.2. Linux Build	4

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). By default the osx and unix build scripts will attempt to download (and patch where applicable) these dependencies from freely available sources. Alternatively, zip files of the fully patched 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.

The latest open source version of the NASA Worldwind JDK is located at <http://worldwind.arc.nasa.gov/java/index.html>, the protolib source at <http://downloads.pf.itd.nrl.navy.mil/protolib/>, and the joglutils source at <https://github.com/sgothel/jogl-utils>. README.joglutils and README.worldwind files are available in the docs directory that detail modifications required to these distributions (Note that these changes are applied automatically by the patch scripts used by the makefiles).

Note that Java 7 is currently required to build sdt3d. Instructions to build with Java 6 available upon request. Git is required in order to download joglutils. wget is used to download worldwind.

1.1. Linux Build

For Linux, makefile are provided that will build either a 32 bit or 64 bit sdt3d application. (You may also build sdt3d under linux via Eclipse as described in the windows build section.)

The makefile will download and patch source dependencies from freely available sites listed above if joglutils.zip, worldwind.zip, and protolib.tgz files do not already exist in the makefiles directory. (If you have access to protean forge you may download the prepatched files from from the sdt file release section at <https://pf.itd.nrl.navy.mil/sf/frs/do/viewSummary/projects.sdt/frs>. Make sure the files are named joglutils.zip, worldwind.zip, and protolib.tgz)

To download and patch the source dependencies the makefile requires the "git", "ant", "g++", and "wget" packages. Note that the optional helper script "prep-dependencies-linux" is available that will use apt-get to download dependencies on linux systems.

Note that you must have JAVA_HOME defined and javac and jar available in your path. When using the openJDK package under Ubuntu you may also need to set java home via update-alternatives, e.g.

```
# set JAVA_HOME to be used by update-alternatives
```

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64 # replace as appropriate to your java openjdk location
update-alternatives --display java
update-alternatives --display javac
update-alternatives --display jar
```

Later Ubuntu distributions require libvecmath-java

```
apt-get install libvecmath-java
```

To build protolib.tgz the pthread and netfilter-queue packages are required:

```
apt-get install libpcap-dev
apt-get install libnetfilter-queue-dev
```

From the makefiles directory make sdt3d. Note that "make -f Makefile.<os> all" will make sdt, sdtcmd, and sdt3d.

```
cd sdt/makefiles
make -f Makefile.linux_<archType> sdt3d
```

If something goes wrong with the above build command it may be helpful to clean out the build directory via:

```
make -f Makefile.linux_<archType> distclean
```

All the files required to run the application are bundled into a sdt3d.zip file located in the makefiles directory. The sdt3d.sh helper script in the makefiles directory can be used to launch the application:

```
./sdt3d.sh
```

The sdtcmd utility can be built via:

```
cd sdt/makefiles
make -f Makefile.linux_<archType> sdtcmd
```

1.2. OSX Build

A makefile is provided that will build a sdt3d application bundle. (You may also build sdt3d under osx via Eclipse as described in the windows build section.)

Note that you must have JAVA_HOME defined and javac and jar available in your path. You may either let the makefile script download and patch the dependent source trees from freely available sites or you may download the latest worldwind and joglutils zip files available in the latest sdt file release directory. "git", "ant", and "wget" will be required if you choose to download and patch the files. From the makefiles directory make sdt3d. A mac application bundle will be created in the makefiles directory. Note that "make -f Makefile.<os> all" will make sdt, sdtcmd, and sdt3d. A sdt3d.zip file will also be created that contains the sdt3d.app and a sdt3d "linux like" directory structure (e.g. the sdt3d.sh script in in the sdt3d directory can be used to launch sdt3d).

On later osx distributions it may be necessary to install vecmath.jar available at <http://jogamp.org/deployment/java3d/1.6.0-pre11>. Copy vecmath.jar to /System/Library/Java/Extensions and set read access to the file.

```
cd sdt/makefiles
make -f Makefile.macosx sdt3d.app
```

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/sdt/trunk"
```

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 library 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 --build-java
waf
```

You may need to set the msvc_targets as appropriate for your platform, e.g. x64, x86,amd:

```
waf configure --build-java --msvc_targets=x86,amd
```

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.

You may need to modify the joglutils java build path to find the java jre (rt.jar) library or the java jdk in the appropriate location for your system. (See the joglutils project properties java build path libraries section).

If you do not have developer access to the protean forge website, you may use the prep-joglutils script within a cygwin environment to download and patch the joglutils freely available source code.

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).

If you do not have developer access to the protean forge website, you may use the prep-wwj script within a cygwin environment to download and patch the freely available worldwind source code.

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/sdt/trunk"
```

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 or makefiles/win64 directories. 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/teantools/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
```