# Building with Ant

I have only tested this with ant build 1.4.1 to 1.5 so do not know if it is backwards compatible with previous versions.

A lot of new options and targets have been added by Christian Geisert. This really makes the build process more flexible and complete. Great thanks!!!! For a list of these see step 7. below.

> **Note:**
>
> The new .jar file will be in the dist directly and named tn5250j and not my5250j. This is for releases 0.5.5 and forwards.

## 1 Getting Ant

Apache Ant is a Java-based build tool and is part of the Apache Jakarta Project

It can be found at the following link:

http://jakarta.apache.org/ant/index.html

## 2 Installing Ant

Please refer to the Ant Documentation

http://jakarta.apache.org/ant/manual/index.html

## 3 Obtaining the tn5250j source

You can obtain the current sources here:

- Sources
- Nightly CVS Tarball
- Anonymous CVS Repository Access

```
cvs -d:pserver:anonymous@cvs.tn5250j.sourceforge.net:/cvsroot/tn5250j login

cvs -z3 -d:pserver:anonymous@cvs.tn5250j.sourceforge.net:/cvsroot/tn5250j
co tn5250j
```

## 4 Steps to get started after setting up ant

1. Create a base directory to place source

2. Create two directories beneath the base directory called lib and src for example your directory structure should look like the following:

**Note:**
As of 0.5.5 the lib directory is created for you during cvs checkout and also provided within the source downloads.

```
base directory
   |
   + - lib
   |
   + - src
```

3. Unzip and place all source files from tn5250j source zip file that was downloaded into src directory.

4. Place the included build.xml file into the base directory.

5. Place the extra .jar files used for mailing in the lib directory.

**Note:**
this is very important or some modules will not compile

You can download the e-mail api's from sun. You will need two packages:

- http://java.sun.com/products/javamail/
- http://java.sun.com/products/javabeans/glasgow/jaf.html

JavaMail jar files:

- activation.jar
- mail.jar
- mailapi.jar
- smtp.jar

**Note:**
Scripting jython.jar file is included in the tn5250j-0.x.x-bin-scripting.zip file and is required as well as of 0.5.5

6. Now execute ant using one of the methods described in the ant documentation (simple typing `ant` should work here) or use the sample bat file for windows below:

```
c:\basedirectory>build all
```

Sample Windows build.bat that I use on windows

```
@echo off

rem this batch-file can be used to override the java_home or ant_home
rem environment or to set a special classpath and so on...
set ANT_HOME=D:\jakarta-ant-1.4.1
set JAVA_HOME=C:\jdk1.3.1

call %ANT_HOME%\bin\ant -buildfile build.xml %1 %2 %3 %4 %5 %6 %7 %8 %9
```

For more information on the command line parameters that the build process supports execute the following:

```
>ant -projecthelp
```

## 5 Building in SSL support

The build process does not include SSL support as default. To compile the files that are associated with SSL you must use the -DSSL=true flag when executing the build process.

> **Warning:**
>
> If you attempt to build in SSL support using version 1.3 you will get errors.

Example

```
    >ant -DSSL=true all
```

A special process will have to be followed to allow the code base to interact with 1.3 and 1.4 version at the same time. This is a temporary solution for now until the build process can test for the different versions.

You will now have a .jar file that works for versions 1.3 and 1.4 of the jdk. All will work except for allocating a Secure Socket if using a version of 1.3. However if the JVM that is executing the emulator happens to be 1.4 then the SSL code will loaded correctly to allocate the Secure Socket.

## 6 Main Build Targets

```
 all                                 Clean, Compile, Jar
 clean                               Cleanup your workspace (delete
build-directory, ...)
 compile                          Compile your source-files
 dist                             Creates all distribution archives
 dist-bin                         Create a Zip and TGZ Archives with all
the binary files
 dist-bin-nosubdir-zip            Create a Zip-File with all the binary
files no sub directory
 dist-bin-prepare                 Prepares binary distribution
 dist-bin-scripting-nosubdir-zip  Create a Zip-File with all the binary
files no sub directory
 dist-bin-tgz                     Create a Tgz-File with all the binary
files
 dist-bin-zip                     Create a Zip-File with all the binary
files
 dist-src                         Create a Zip and TGZ Archives with all
source files
 dist-src-prepare                 Prepares source distribution
 dist-src-tgz                     Create a Tgz-File with all the source
files
 dist-src-zip                     Create a Zip-File with all the source
files
 info                             Prints a short summary of this project
 javadoc                          Creates the javadoc API documentation
 package                          Create a Jar-File of your compiled
classes
```